РЕФЕРАТ

Данная пояснительная записка содержит 79 стр., 26 рисунков, 10 таблиц, 21 источник, 6 приложений.

Ключевые слова: нейронные сети, выборка, обучение, прогноз, временные ряды, стационарность.

Цель работы: Разработать и реализовать алгоритм предварительной обработки данных и последующее выполнение прогноза поведения временных рядов.

Объектом исследования является прогнозирование движения показателей объемов добычи угля.

Предмет исследования – временные ряды, набор данных, который лежит в основе построения модели.

Актуальность проекта обусловлена стремлением промышленных предприятий проводить грамотную экономическую политику, экономить и оптимизировать ресурсопотребление предприятия.

СОДЕРЖАНИЕ

Перечень обозначений и сокращений

ПО – программное обеспечение,

АО «СУЭК» – акционерное общество «Сибирская угольная энергетическая компания»,

НС – нейронная сеть,

РФ – Российская Федерация,

MS – Microsoft,

RNN – Recurrent neural network,

LSTM – Long short-term memory,

GRU – Gated Recurrent Unit.

ВВЕДЕНИЕ

На протяжении всего времени, в течение которого людям приходилось принимать различные решения на основе данных, какой бы природы они не были, исследователями этих данных определялись закономерности и стратегии, придерживаясь которых, ими могли быть сделаны определенные выводы, влияющие на дальнейшие действия, такие как планирование и прогнозирование. Прогнозирование является важным инструментом для принятия решений и планирования в различных областях. Ниже перечислены области и причины, по которым людьми были исследованы и структурированы механики и способы получения прогнозов:

1) Предсказание будущих трендов и событий. Прогнозирование позволяет людям предсказывать изменения в экономике, политике, технологиях и других сферах. Это помогает бизнесам, правительствам и организациям адаптироваться к изменениям и принимать соответствующие решения.

2) Бюджетирование и планирование. Прогнозирование помогает людям и организациям планировать свои финансы, ресурсы и временные рамки. Например, для бизнесов это важно при разработке бюджета на следующий год или планировании производственных циклов.

3) Улучшение стратегий и принятие решений. Прогнозирование позволяет анализировать различные сценарии и выбирать оптимальные стратегии. Например, в маркетинге прогнозы могут помочь определить наиболее эффективные каналы продаж или рекламные кампании [1].

В настоящее время прогнозирование экономических процессов является неотъемлемой частью ведения бизнеса. Что привело к необходимости внедрения инструментов прогнозирования в интеллектуальный анализ? Прогнозирование экономических процессов позволяет более точно оценить будущую ситуацию и принять обоснованные, актуальные для управляющего аппарата решения. Это особенно важно для государственных органов, банков, предприятий и инвесторов, которым необходимо планировать свою

деятельность, а также для семей и индивидуальных лиц, которые хотят оптимизировать свои расходы и инвестиции. Успешные прогнозы могут помочь принимать правильные инвестиционные решения с минимизацией рисков. Прогноз доходов необходим компании не для определения будущих финансовых показателей, а для разработки стратегии и тактики на прогнозный период [2].

В данной работе будет спроектирована и реализована модель нейронной сети для предоставления прогнозов на основе временного ряда.

Основными задачами данной работы являются:

1) Провести анализ предметной области, рассмотреть и сравнить аналоги;
2) Определить требования и ограничения информационной системы, объекта исследования и модели;
3) Провести сбор статистической информации и сформировать данные;
4) Разработка модели программного инструмента нейронной сети;
5) Тестирование модели нейронной сети;
6) Анализ полученных результатов.

1 Анализ предметной области

1.1 Основные понятия анализа временных рядов

Под временным (динамическим) рядом понимают последовательность наблюдений некоторого признака X (случайной величины) в последовательные моменты времени t. Уровнями ряда называются отдельные наблюдения, которые обозначаются x_t, t = 1,..,n.

Существуют 3 основных типа временных рядов:

1. Аддитивная модель, в которой в которой компоненты суммируются.
2. Мультипликативная модель, в которой компоненты перемножается.
3. Смешанная модель.

Выбор одной из трех моделей осуществляется на основе анализа структуры сезонных колебаний. Если амплитуда колебаний приблизительно постоянна, строят аддитивную модель временного ряда, в которой значения сезонной компоненты предполагаются постоянными для различных циклов. Если амплитуда сезонных колебаний возрастает или уменьшается, строят мультипликативную модель временного ряда, которая ставит уровни ряда в зависимость от значений сезонной компоненты.

При исследовании временного ряда выделяют несколько составляющих:

$$x_t = u_t + \gamma_t + c_t + e_t,\ t = 1 \dots n,$$

где u_t – тренд,

γ_t – компонента сезонности, отражающая повторяющуюся динамику процесса в рамках какого-либо повторяющегося периода, обычно меньшего чем год,

c_t – циклическая компонента, которая отражает глобальную сезонность, ставшую цикличностью. Применимо в большинстве случаев для периодов длительностью год и более,

e_t – случайная (стохастическая) компонента, являющаяся результатом воздействия на исследуемый процесс случайных факторов.

Временные ряды описываются по следующим признакам:

− по форме представления уровней: ряды абсолютных показателей, относительных показателей, средних величин;

− по количеству показателей: одномерные и многомерные;

− по характеру временного параметра: моментные и интервальные;

− по полноте данных: полные и неполные;

− детерминированные и случайные;

− по наличию основной тенденции: стационарные и нестационарные ряды. Можно выделить следующие этапы анализа и прогнозирования временных рядов:

− графическое представление и анализ составляющих ряда;

− проверка ряда на стационарность;

− построение модели прогнозирования;

− прогнозирование временного ряда на основе ранее проведенных исследований [3].

Необходимо определить является ли исследуемый временной ряд x_t, t = 1,..,n называется строго стационарным, если совместное распределение вероятностей n наблюдений x_1, x_2,…, $x_{n+\tau}$ при любых n, t, τ. То есть у стационарных временных рядов вероятностные характеристики не зависят от момента t. Поэтому математическое ожидание $M(x_t) = a$ и среднеквадратическое отклонение могут быть оценены по значениям x_t по формулам:

$$a = \frac{1}{N}\sum_{i-1}^{N} x_i(t_i),$$

$$\sigma = \sqrt{\frac{\sum x_i^2}{n} - \bar{x}^2}.$$

Из стационарности временного ряда следует, что такие свойства, как дисперсия, математическое ожидание и ковариация неизменны со временем.

Для того, чтобы проверить временной ряд на стационарность, необходимо провести тест Дикки-Фуллера, который проверяет нулевую гипотезу о наличии единичного корня.

Таким образом, если наблюдаемое значение критерия больше 0,05, то не удается отклонить нулевую гипотезу, следовательно, существует единичный корень и ряд является нестационарным.

Если же значение меньше 0.05, то нулевая гипотеза отклоняется, и ряд является стационарным [4].

1.2 Описание структуры и объема данных

В качестве набора данных были выбраны и сформированы списки показателей помесячной выработки объемов угля энергетического в бассейнах Российской Федерации в период с января 2018 г. по декабрь 2023 г. АО «Сибирская угольная энергетическая компания».

АО «СУЭК» - ведущий продуцент угля в России, характеризующийся устойчивым ростом объемов добычи и широким покрытием угольных бассейнов по всей стране от Кузбасса до Дальнего Востока. Доля угольного сырья, добытого компанией, обеспечивает около четверти отечественного угольного производства. По итогам марта 2024 года доля составляет 21,8 %.

Под энергетическим углем следует понимать весь уголь, объем которого удовлетворяет требованиям и запросам энергетического сектора страны, усваивается на внутреннем рынке. В данной работе использовались показатели выработки данной разновидности угля. Такой вид сырья был выбран не случайно. Угольная промышленность играет важнейшую роль в экономике России, обеспечивая значительный вклад в энергетический сектор и промышленность государства, а также оказывает положительное влияние на региональное развитие.

По итогам сбора информации и формирования набора данных было сформировано 73 строки данных показателей добычи угля предприятием АО «СУЭК».

В ходе анализа данных и дальнейшего прогнозирования будут построены графики для демонстрации поведения исследуемого временного ряда.

1.3 Анализ существующих программных решений для реализации производственного прогнозирования

В настоящее время существует несколько популярных технологий, решающих поставленную задачу и частично или полностью поддерживающие русский язык: Loginom, KNIME Analytics, SAS Enterprise, Форсайт.

Loginom представляет собой программный продукт от компании Loginom company и предназначен для анализа и обработки бизнес-данных на базе методов визуального проектирования, является универсальным конструктором с набором готовых компонентов. Делает продвинутую аналитику доступной конечным пользователям без привлечения IT-специалистов, позволяя автоматизировать бизнес-процессы и кратно ускорить работу с данными. Данный инструмент входит реестр программного обеспечения Российской Федерации, что делает Loginom привлекательным для предприятий государственного сектора. Данный сервис требует значительную плату за редакцию для профессиональной аналитики, вводит ограничения на функционал для некорпоративного пользователя. С использованием облачного API данный инструмент имеет возможности интеграции и разворачивания ИС на Яндекс Cloud.

KNIME Analytics Platform – это программная платформа анализа, интеграции данных и подготовки отчётности с открытым исходным кодом. Аналитическая платформа KNIME предоставляет пользователям возможности визуально создавать потоки данных (конвейеры), выборочно выполнять отдельные или все шаги анализа, а затем проверять результаты, модели, используя интерактивные виджеты и представления. Однако данный инструмент не имеет поддержки иных языков кроме английского. KNIME не является бесплатным, имея разные уровни тарификации в зависимости от требований, тарифы образуются в зависимости от валюты Евро, что делает стоимость тарифов нестабильной и завышенной для российских компаний. Также для KNIME реализованы API возможности.

Программный продукт SAS Enterprise Miner - это интегрированный компонент системы SAS, созданный специально для выявления в огромных массивах данных информации, необходимой для принятия решений. Разработанный для поиска и анализа глубоко скрытых закономерностей в данных SAS, Enterprise Miner включает в себя методы статистического анализа, соответствующую методологию выполнения проектов Data Mining (SEMMA) и графический интерфейс пользователя. Данный инструмент не имеет поддержки иных языков кроме английского, а также не подходит для некоммерческого любительского пользования. Тарификация функционала SAS Enterprise производится по запросу и формируется из запрашиваемого функционала и требуемых мощностей, что делает инструмент непривлекательным в финансовом плане.

Форсайт. Аналитическая платформа – это программный комплекс для интеллектуального анализа данных, позволяющий эффективно визуализировать информацию для обеспечения принятия бизнес-решений на основе надёжных данных. Данный инструмент имеет индивидуализированную тарификацию и предназначен, большим образом, для корпораций, что делает инструмент непривлекательным и непопулярным в лице некоммерческого обывателя.

В таблице 1 приведены ключевые характеристики сравниваемых технологий: страна производитель инструмента, кроссплатформенность, поддержка API для интеграции в свои сервисы, возможность использования инструмента в личных некорпоративных целях.

Таблица 1. Сравнительный анализ программных решений

Технология	Страна разработчик	Кроссплатформенность	API	Возможность использования в личных некорпоративных целях
Loginom	Россия	-	+	+
KNIME Analytics	Швейцария	+	+	+
SAS Enterprise	США	-	-	-
Форсайт	Россия	+	+	-

По результатам формирования таблицы 1 можно сделать вывод, что наиболее лучшими решениями в настоящее время являются KNIME Analytics и Форсайт, не смотря на одинаковые характеристики с Loginom. Все потому, что аккредитованный инструмент Loginom подходит для пользования только внутри отечественных операционных систем, что делает его негибким и неудобным для реализации в настоящее время.

Исходя из анализа конкурентов можно выделить следующие причины, по которым разработка собственной системы может быть оправдана:

1) Не все перечисленные инструменты имеют поддержку взаимодействия с российскими облачными технологиями;

2) Для российского рынка нет однозначно кроссплатформенного и адаптированного для русскоязычного аналитика/разработчика программного решения;

3) Среди перечисленных инструментов нет предложений с уровнем ценообразования оптимальным для частного некорпоративного пользования.

1.4 Обзор применяемого инструментария

Для разработки ИС будут использованы следующие средства: Python — это язык программирования, который будет использоваться для разработки всего алгоритма. Он обладает высокой читаемостью, простотой использования и обширной библиотекой модулей, что делает его идеальным выбором для данного проекта. По сравнению с другими языками программирования он может обеспечить простую и быструю разработку нейронных сетей.

Для оптимизации работы с реализуемыми алгоритмами была выбрана интерактивная облачная среда Google Collab. Google Collab — сервис, созданный Google, который предоставляет возможность работать с кодом на языке Python через Jupyter Notebook, не устанавливая на свой компьютер дополнительных программ. В Google Collab можно применять различные библиотеки на Python, загружать и запускать файлы, анализировать данные и получать результаты в браузере.

Keras — это высокоуровневая библиотека API для Tensorflow, которая упрощает разработку нейронных сетей. Она позволяет создавать модели нейронных сетей с помощью простых и понятных функций, что делает ее идеальным выбором для данного проекта.

Matplotlib — это библиотека для визуализации данных, которая будет использоваться для визуализации результатов работы нейронной сети. Она позволяет создавать различные типы графиков и диаграмм, что помогает в анализе и интерпретации данных.

Numpy — это библиотека для работы с многомерными массивами данных, которая будет использоваться для обработки изображений. Она позволяет выполнять различные операции над массивами данных, что необходимо для подготовки изображений к распознаванию.

Scikit-learn (sklearn) — это один из наиболее широко используемых пакетов Python для Data Science и Machine Learning. Он содержит функции и

алгоритмы для машинного обучения: классификации, прогнозирования или разбиения данных на группы.

SciPy — это библиотека для языка Python, основанная на расширении NumPy, но для более глубоких и сложных научных вычислений, анализа данных и построения графиков. SciPy в основном написана на Python и частично на языках C, C++ и Fortran, поэтому отличается высокой производительностью и скоростью работы. Библиотека необходима для машинного обучения и создания моделей искусственного интеллекта, прогнозирования и построения моделей.

Pandas — это библиотека Python для обработки и анализа структурированных данных, её название происходит от «panel data» («панельные данные»). Панельными данными называют информацию, полученную в результате исследований и структурированную в виде таблиц. Для работы с такими массивами данных и создан Pandas [5].

2 Формирование и предобработка данных

2.1 Поиск и сбор данных

Для моделирования процесса прогнозирования прибыли производственного предприятия необходимо определить, изъять и стандартизировать предшествующие исследуемому периоду производственные показатели предприятия АО «СУЭК». В соответствии с приказом Минэкономразвития России от 24 мая 2021 г. № 279 «Об утверждении Порядка утверждения Федеральной службой государственной статистики форм федерального статистического наблюдения и указаний по их заполнению» источником необходимых производственных и макроэкономических показателей послужила Федеральная служба государственной статистики. В процессе сбора информации были обнаружены сложности с форматированием требуемых данных. Вышеописанный источник информации приводил ее в форме отчетности формата PDF и PPTX за узкий период, в следствие чего идея об автоматизированном сборе информации была отброшена в пользу ручной самостоятельной структуризации.

Были выделены и перечислены следующие, необходимые для моделирования, показатели:

1) Средняя стоимость 1 тонны энергетического угля РФ (в рублях);
2) Объем добычи энергетического угля РФ (в млн. тонн);
3) Объем добычи энергетического угля АО «СУЭК» по месяцам (в тыс. тонн);
4) Коэффициент годовой инфляции РФ (в %).

В целях уточнения приведенных выше характеристик были сформированы и направленны электронные письма в отделы статистического учета данных АО «СУЭК» и Министерство угольной промышленности Кузбасса. Содержание обращений с указанием адресатов, соответствующих вышеперечисленным ведомствам представлены в Приложении Г.

В процессе формирования выборки для модели прогнозирования была составлена таблица показателей добычи угля АО «СУЭК», содержащая 72 записи объемов добычи в тысячах тонн в период с 01.01.2018 по 01.11.2023 с шагом в 1 месяц. Собранные данные представлены в таблице 2.

Таблица 2.

Добыча угля АО «СУЭК» с 01.01.2018 – 01.11.2023 в тыс.тонн.

Дата	Объем добычи, тыс. тонн	Дата	Объем добычи, тыс. тонн	Дата	Объем добычи, тыс. тонн
01.01.2018	8,20	01.12.2019	8,96	01.12.2021	8,44
01.02.2018	6,61	01.01.2020	7,96	01.01.2022	7,70
01.03.2018	7,19	01.02.2020	6,65	01.02.2022	7,83
01.04.2018	7,87	01.03.2020	7,80	01.03.2022	8,09
01.05.2018	6,98	01.04.2020	8,04	01.04.2022	7,43
01.06.2018	7,04	01.05.2020	7,37	01.05.2022	7,17
01.07.2018	8,35	01.06.2020	7,46	01.06.2022	7,17
01.08.2018	10,16	01.07.2020	6,87	01.07.2022	7,13
01.09.2018	10,33	01.08.2020	7,48	01.08.2022	7,59
01.10.2018	8,81	01.09.2020	6,71	01.09.2022	7,76
01.11.2018	8,13	01.10.2020	7,50	01.10.2022	8,63
01.12.2018	6,69	01.11.2020	6,89	01.11.2022	8,72
01.01.2019	6,82	01.12.2020	6,91	01.12.2022	9,44
01.02.2019	10,05	01.01.2021	7,50	01.01.2023	7,61
01.03.2019	6,76	01.02.2021	7,83	01.02.2023	7,67
01.04.2019	7,02	01.03.2021	8,28	01.03.2023	7,96
01.05.2019	10,16	01.04.2021	7,70	01.04.2023	8,00
01.06.2019	7,63	01.05.2021	7,46	01.05.2023	7,83

01.07.2019	8,35	01.06.2021	7,09	01.06.2023	7,28
01.08.2019	7,61	01.07.2021	7,52	01.07.2023	7,39
01.09.2019	7,32	01.08.2021	7,61	01.08.2023	7,43
01.10.2019	8,48	01.09.2021	7,80	01.09.2023	7,63
01.11.2019	7,41	01.10.2021	8,52	01.10.2023	8,48
01.01.2018	8,20	01.11.2021	8,52	01.11.2023	8,57

По показателям, представленным в таблице 2 был сформирован график временного ряда объемов добычи угля, на котором оси абсцисс соответствуют параметры даты, а по оси ординат отложены объемы выработки. График представлен на рисунке 1.

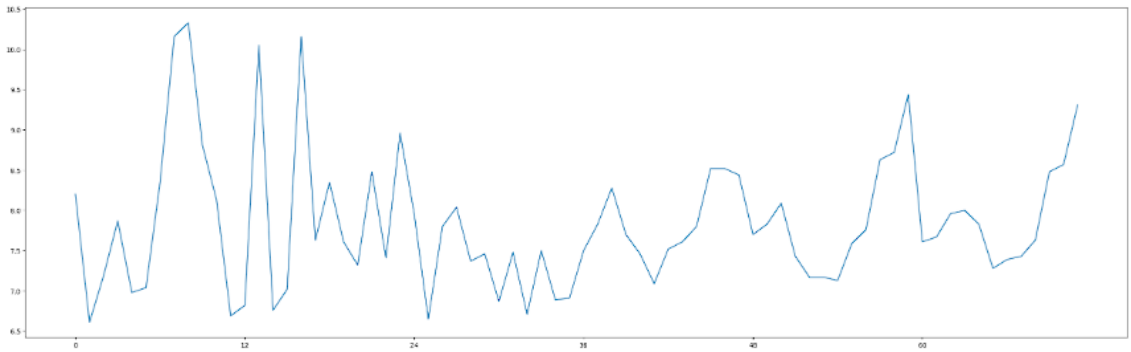


Рисунок 1 – Временной ряд помесячных объемов добычи угля АО «СУЭК»

График частоты возникновения значений выборки сформирован в виде гистограммы и представлен на рисунке 2.

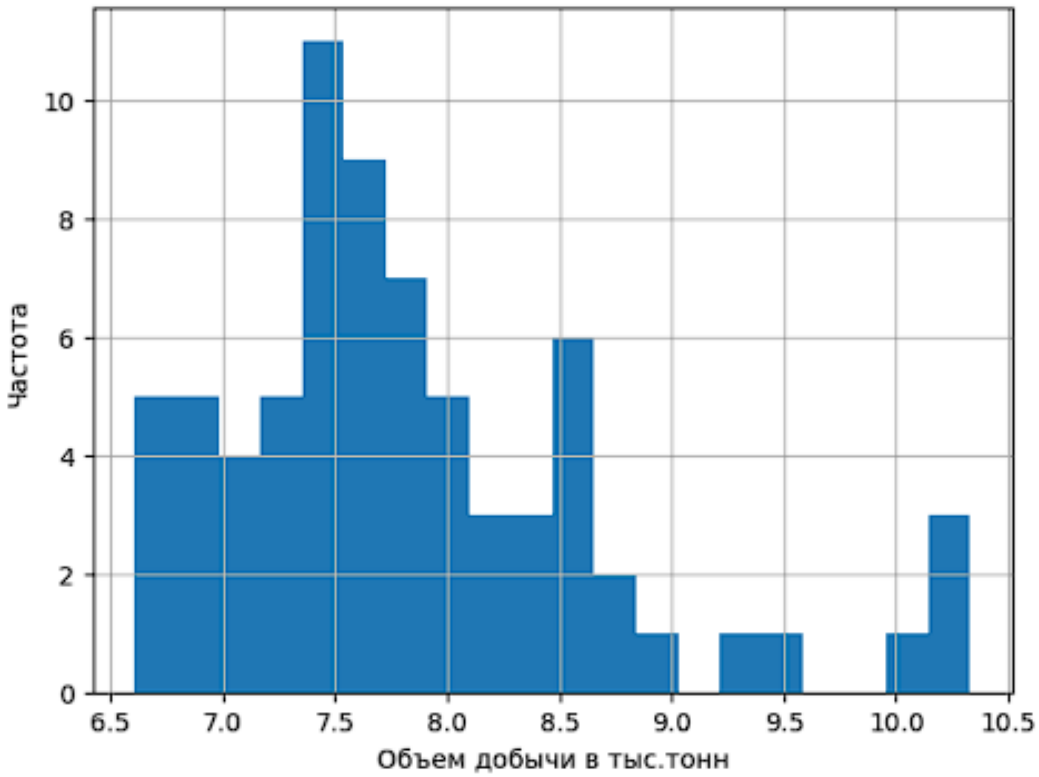


Рисунок 2 – Гистограмма частоты значений исследуемой выборки

По результатам гистограммы, представленной на рисунке 2 можно отметить, что объемы добычи энергетического угля АО «СУЭК» редко превышали 9 тысяч тонн в месяц.

2.2 Анализ экономических показателей

Инфляция представляет собой макроэкономический показатель, отражающий общее увеличение уровня цен на товары и услуги в экономике. Этот показатель измеряется с помощью индекса цен, который отражает изменение цены на корзину товаров и услуг в заданном периоде. Рост инфляции может быть вызван различными причинами, такими как увеличение денежного предложения, рост стоимости сырья, повышение налогов или спроса на товары, изменение характера проводимой страной внутренней и внешней политики. Инфляция оказывает существенное влияние на экономику государства, влияя на покупательскую способность населения, ставки банков. Рост коэффициента инфляции стимулирует экспорт товаров, однако в то же время обесценивает доходы населения. Регуляризация и контроль динамики коэффициента инфляции проводится с помощью денежно-кредитной

политики Центрального банка. Данные годовой инфляции в исследуемом периоде были предоставлены Федеральной службой государственной статистики.

При расчёте прибыли от прогнозируемого объема добычи угольного сырья предприятием нельзя учитывать лишь его себестоимость. Также является недопустимым примерное определение инфляции как инфляционного ожидания – гипотетического значения инфляции, основанного на чувствах и настроениях населения страны, предположение.

Необходимо определить и рассчитать гипотетическую инфляцию на прогнозируемый период для получения наиболее уточненной, приближенной к реальности, стоимости единицы товара (в рамках ВКР – актуализированную себестоимость одной тонны угля на 2024 год) с учтенным макроэкономическим фактором инфляции. Индекс годовой инфляции в период с 2018 по 2023 год представлен в таблице 3.

Таблица 3. Индекс годовой инфляции с 2018 по 2023 год.

Расчётный год	Темп инфляции, %
2018	3,24
2019	3,04
2020	4,91
2021	8,39
2022	11,94
2023	7,42

Отразим наглядную динамику инфляции Российской Федерации. Графическое представление таблицы 3 как точечной диаграммы представлено на рисунке 3.

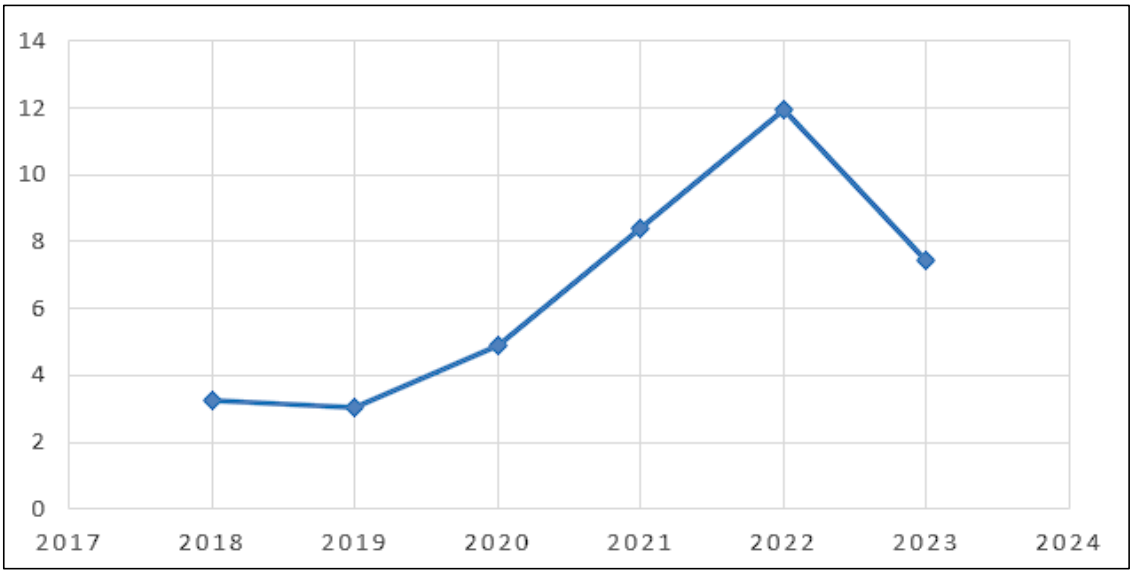


Рисунок 3 – Динамика коэффициента инфляции РФ 2018 – 2023гг.

По результатам диаграммы можно сделать вывод об усилении контроля над темпом инфляции аппаратом Центрального банка РФ. Наблюдается стабильный рост инфляции в период с 2019 года по 2022 год, вызванный мировой пандемией COVID – 19, оказавшей влияние на темп экономического роста страны. Можно выдвинуть промежуточную гипотезу о ниспадающем направлении вектора коэффициента инфляции после резкого скачка в 2022 году, вызванного внутренними и внешнеполитическими факторами.

Пускай индекс цен I за 2018 год будет обозначен как I_1. Тогда последующие индексы, относящиеся к расчетным годам от 2019 по 2023 таблицы 3 градируются как I_2, I_3, … I_6 соответственно.

Определим индекс инфляции I_7 на прогнозируемый период, а именно на 2024 год. Для данного формата была выбрана методология поиска среднегодовой инфляции. Формула расчёта индекса инфляции при различных значениях темпа инфляции за n лет имеет вид

$$I_{n+1} = (r_1 + 1)(r_2 + 1)\ldots(r_n + 1) = I_1 \cdot I_2 \cdot \ldots \cdot I_n \,,$$

где r_i – темп инфляции.

Таким образом, на основе вышеописанной функции можем произвести вычисления, результат которых будет соответствовать темпу инфляции за

2024 год r_7, эквивалентом которого считаем среднегодовой темп инфляции за весь временной промежуток.

Формула расчёта среднегодового темпа инфляции имеет вид:

$$r_{n+1} = \left(\sqrt[n]{I_{n+1}} - 1\right) \cdot 100\% .$$

Был получен параметр темпа инфляции, эквивалентный прогнозируемому темпу инфляции на 2024 год и равен 12,87 %.

Чтобы удостоверится в адекватности полученного параметра инфляции было принято решение прибегнуть к функционалу инструмента «Лист прогноза» MS Excel. Это инструмент для анализа данных, который позволяет выделить набор изменяющихся во времени данных и спрогнозировать их дальнейшее изменение. При этом Лист прогноза позволяет установить прогнозируемое значение в виде интервала, крайние значения которого обозначаются как верхняя и нижняя доверительные границы. На рисунке 4 представлено графическое отображение прогноза данным методом, где синей кривой графика соответствуют фактические значения годовой инфляции 2018 – 2023 гг., а оранжевым обозначена линия прогноза, тонкой оранжевой кривой - нижняя и верхняя доверительные границы.

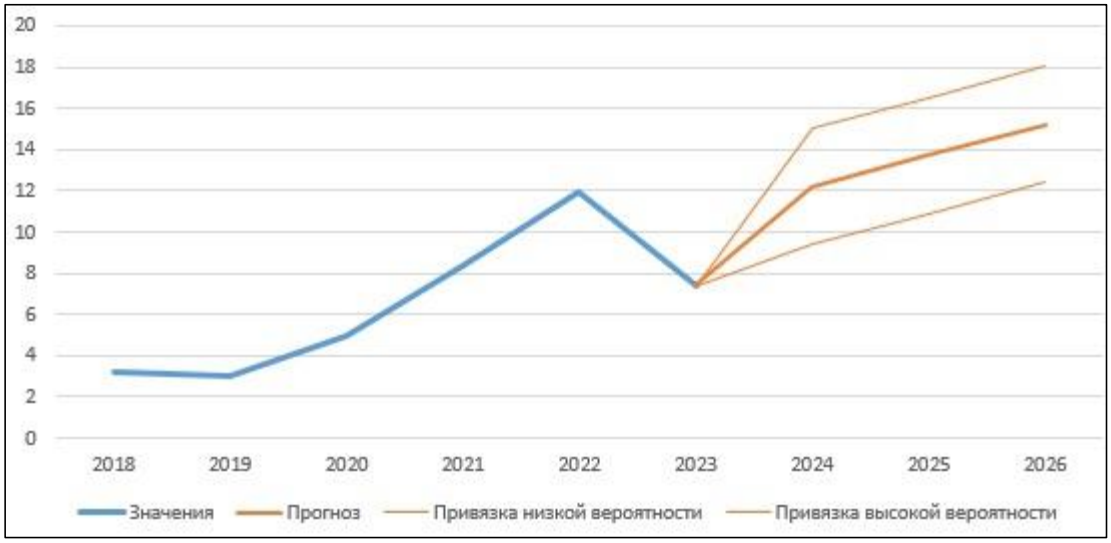


Рисунок 4 – График движения параметра годовой инфляции методом Листа прогноза MS Excel

По результатам метода имеем параметры инфляции и граничных точек разброса этого значения на 2024 год, представленные в таблице 4.

Таблица 4 – Результаты прогнозирования инфляции на 2024 год методом «Лист прогноза» MS Excel

Прогноз инфляции, %	Нижняя граница, %	Верхняя граница, %
12,21%	9,38%	15,04%

Таким образом, среднегодовая инфляция, найденная по формулам и равная 12,87 % в большей степени совпадает с темпом инфляции в 12,21 %, предоставленной Листом прогноза. В следствии, можно сделать вывод о том, что параметр инфляции I_7 на 2024 год равный 12,87 % в рамках ВКР принято считать приемлемым.

Как любой товар или услуга уголь имеет цену, в которую закладывается:

- стоимость его добычи,
- логистические расходы
- стоимость труда, необходимого для выработки,
- организация и реализации процессов добычи, транспортировки, усвоения.

Торговой углеродной единицей в отношении Российского угля принято считать одну тонну. Таким образом в период с 2019 по 2023 год стоимость угля изменялась образом, представленным в таблице 5.

Таблица 5. Средняя стоимость 1 тонны энергетического угля РФ 2019-2023 гг.

Расчётный год	Стоимость, рубль/тонна
2019	3365
2020	2517
2021	3072
2022	1914
2023	2977

Обозначим переменную стоимости как С. Тогда стоимость одной тонны угля на 2024 год, то есть прогнозируемую, обозначим C_{2024} и рассчитаем, как

$$C_{2024} = C_{2023} \cdot (r_7 + 1)$$

В результате приведенных вычислений прогнозируемая стоимость одной тонны энергетического угля в 2024 году составляет 3360 (три тысячи триста шестьдесят) рублей/тонна.

2.3 Стандартизация и предобработка данных

Предварительно, перед формированием обучающих и тестовых наборов данных, которые будут подаваться на вход модели производится импорт данных и выделение необходимых в работе атрибутов базы данных, а именно атрибуты Date и Coal, значения даты и соответствующего ей объема добытого угля в тысячах тонн.

Для обучения нейронных сетей используются данные формата CSV (Comma Separated Values). Формат CSV это текстовый формат для представления табличных данных. Каждая строка в файле описанного формата представляет отдельную запись или строку, состоящую из отдельных столбцов, разделенных запятыми, как следует из перевода расшифровки аббревиатуры CSV с английского языка на русский.

Так как формирование данных производилось инструментарием MS Excel, следовательно, предварительно данные хранились в формате .xlxs, который не подходит для корректного отображения содержимого датасета, экспортированного в среду разработки Google Collab на базе Jupiter Notebook. MS Excel предоставляет возможность преобразования набора данных в формат CSV с разделителем запятой, что ускоряет стандартизацию набора и позволяет избежать ручной установки разделителей и формировании строк через текстовые редакторы, например, приложение Блокнот, позволяющее получить при сохранении файл формата CSV с разделителями, назначенными вручную. Такой метод нельзя назвать надежным в силу невозможности исключения ошибки человеческого фактора для процесса ручного ввода

данных и разделителей. Вид 20-ти строк данных в формате CSV представлен на рисунке 5.

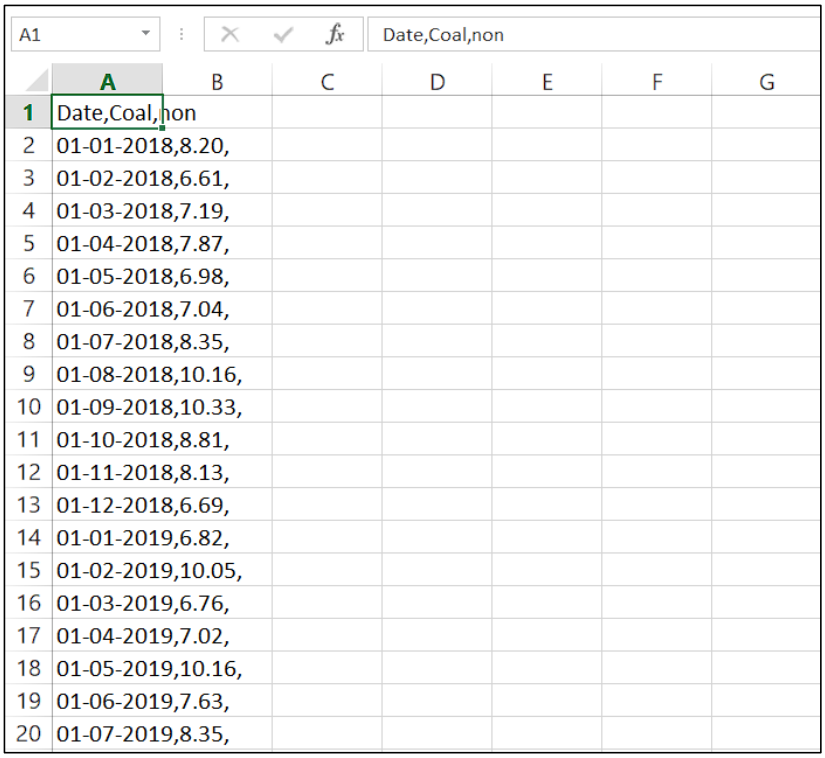


Рисунок 5 – Представление набора данных о добыче угля в формате CSV

2.4 Статистический анализ данных временного ряда

Автокорреляция уровней временного ряда – корреляционная зависимость между последовательными уровнями временного ряда. Поиск автокорреляции позволяет отметить в наличии исследуемого временного ряда тенденции и циклические колебания, при выявлении которых можно дать оценку зависимости будущих значений от предшествующих.

Порядок коэффициента автокорреляции определяет сдвиг на определённый промежуток времени L – лаг. При $L = 1$ имеем коэффициент автокорреляции первого порядка. Необходимо учитывать, что при увеличении лага на единицу количество пар значений, необходимых для расчёта автокорреляции, уменьшается на 1. Максимальный порядок коэффициента автокорреляции рекомендуют такой, чтобы индекс последнего порядка не превышал $n/4$, где n – количество элементов исследуемого ряда.

Вычисление коэффициентов автокорреляции позволяет выявить структуру ряда и определить лаг, при котором автокорреляция $r_{t, t-L}$ наиболее высокая. Если наибольшим значением оказывается значение $r_{t,\ t-L}$, то ряд содержит и тенденцию, и колебания периодом L. Если наиболее высоким оказывается значение $r_{t,\ t-1}$, то такой временной ряд содержит линейную тенденцию, но при этом не исключена возможность существования нелинейной тенденции.

Если ни одно из двух значений не является значимым, то можно выдвинуть одно из двух предположений о структуре временного ряда:

– ряд не содержит тенденции и циклических колебаний, а его уровень определяется только случайной компонентой;

– либо ряд содержит сильную нелинейную тенденцию, для выявления которой необходим дополнительный анализ.

Автокорреляционной функцией временного ряда называют последовательность коэффициентов автокорреляции 1, 2 и т. д. порядков.

Коррелограмма – график зависимости значений коэффициентов автокорреляции от величины L (порядка коэффициента автокорреляции).

2.4.1 Автокорреляция уровней временного ряда. Расчет коэффициента автокорреляции первого порядка.

Приведем таблицу с данными, отражающими динамику изменения объемов добычи угля в период с 2018 по 2023 год, т.е. шестилетний период по месяцам, итого временной ряд из 72 значений. Для получения корректных показателей автокорреляции минимальное количество коэффициентов должно быть не меньше шести. В свою очередь, для данного ряда данных приведем вычисления коэффициентов автокорреляции для 8-ми порядков. Месяц обозначим как переменную t в условных единицах. Данные для которых исследуется автокорреляция уровней ряда и расчет автокорреляции сформированы в таблице 2.

Ниже приведен подробный алгоритм расчета автокорреляции 1-го порядка в общем виде.

Для данного временного ряда рассчитаем автокорреляции с 1-го по 8-ий порядок.

1) Вычисляются выборочные средние:

$$\bar{x} = \frac{\sum x_i}{n}$$

$$\bar{y} = \frac{\sum y_i}{n}$$

$$\overline{xy} = \frac{\sum x_i y_i}{n}$$

2) На основе значений, полученных пунктом 1, вычисляются выборочные дисперсии:

$$S(x)^2 = \frac{\sum x_i^2}{n} - \bar{x}^2$$

$$S(y)^2 = \frac{\sum y_i^2}{n} - \bar{y}^2$$

3) Из полученных значений дисперсии для последовательностей x_i и y_i производится расчет среднеквадратического отклонения:

$$S(x) = \sqrt{S^2(x)}$$

$$S(y) = \sqrt{S^2(y)}$$

4) Производится расчет линейного коэффициента автокорреляции $r_{t,\,t-1}$:

$$r_{t,t-1} = \frac{\overline{x_t \cdot x_{t-1}} - \bar{x_t} \cdot \overline{x_{t-1}}}{S(x_t) \cdot S(x_{t-1})}$$

Существует способ автоматизации поиска значений автокорреляционной функции и коррелограммы для временных рядов, содержащих множество значений путем подключения узкоспециализированных надстроек MS Excel, предназначенных для проведения оценки значимости отклонений от нормальности для временного ряда, проведение автокорреляционного анализа и визуализации результатов и зависимостей.

Также для временных рядов двух зависимых переменных можно использовать функцию MS Excel «КОРЕЛЛ()» из базового инструментария

программы. Данная функция возвращает коэффициент корреляции двух диапазонов ячеек. С помощью полученного значения корреляции можно определить силу и род взаимосвязи между переменными: например, можно установить зависимость между возрастом домашней кошки и среднесуточным расстоянием, которое эта кошка проходит.

Установленный в среду MS Excel инструмент «ACF.HCXL» проводит автокорреляционный анализ временного ряда, вычисляет оценочные значения для него и сводит полученные результаты в формате таблиц на отдельный лист MS Excel автоматически. Лист, по умолчанию, будет назван по аббревиатуре самой надстройки: «ACF.HCXL». Описанный инструмент является наиболее подходящим в исследовании рассматриваемого в ВКР бизнес-процесса. Данный вывод был сделан на основе факта, что сформированный временной ряд объемов производства угля не имеет зависимых от него переменных и, в свое время, не является зависимым от какой-либо переменной. На рисунке 6 представим общий вид листа результатов работы надстройки для проведения автокорреляционного анализа временного ряда.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Корреляционный анализ ряда остатков прогноза ACF HCXL											
2												
3	n	Ряд остатков X(t) = Y(t)-F(t)	Оценки значимости отклонений от нормальности				АКФ(...)	ЧАКФ(...)	Ошибка АКФ		Ошибка ЧАКФ	
4	1	8,20	Отклонение среднего значения для ряда данных от нуля			1	0,261	0,261	0,270	-0,270	0,272	-0,272
5	2	6,61	*Среднее значение для данных <X> =*	7,827		2	-0,016	-0,090	**0,288**	**-0,288**	**0,276**	**-0,276**
6	3	7,19	*крит. значение <x> = 2s/√n*	0,197		3	0,069	0,105	**0,288**	**-0,288**	**0,281**	**-0,281**
7	4	7,87	*Стандартное отклонение для данных <s> =*	0,834		4	-0,206	-0,280	**0,289**	**-0,289**	**0,304**	**-0,304**
8	5	6,98				5	-0,068	0,102	**0,300**	**-0,300**	**0,309**	**-0,309**
9	6	7,04	Отклонения долей полож. и отриц. остатков от 50%			6	-0,110	-0,201	**0,301**	**-0,301**	**0,322**	**-0,322**
10	7	8,35	*ΔN± =|n₊ - n₋| /n =*	1,000		7	-0,092	0,079	**0,304**	**-0,304**	**0,326**	**-0,326**
11	8	10,16	*крит. значение ΔN± = 2,4/√n*	0,236		8	0,087	0,000	**0,306**	**-0,306**	**0,328**	**-0,328**
12	9	10,33										
13	10	8,81	Автокорреляций: Q-статистика Льюинга-Бокса									
14	11	8,13	*QLB =*	11,506								
15	12	6,69	*Q крит. =*	15,507								
16	13	6,82										

Рисунок 6 – Результат проведения автокорреляционного анализа надстройки MS Excel «ACF.HCXL»

Приведем сводную таблицу 7 зависимости результатов вычисления автокорреляции от порядка (лага).

Таблица 7. Автокорреляция уровней временного ряда

Порядок (лаг)	$r_{t,\ t-1}$
1	0,261
2	-0,016
3	0,069
4	-0,206
5	-0,068
6	-0,110
7	-0,092
8	0,087

На рисунке 7 представлена коррелограмма автокорреляции для исследуемого временного ряда.

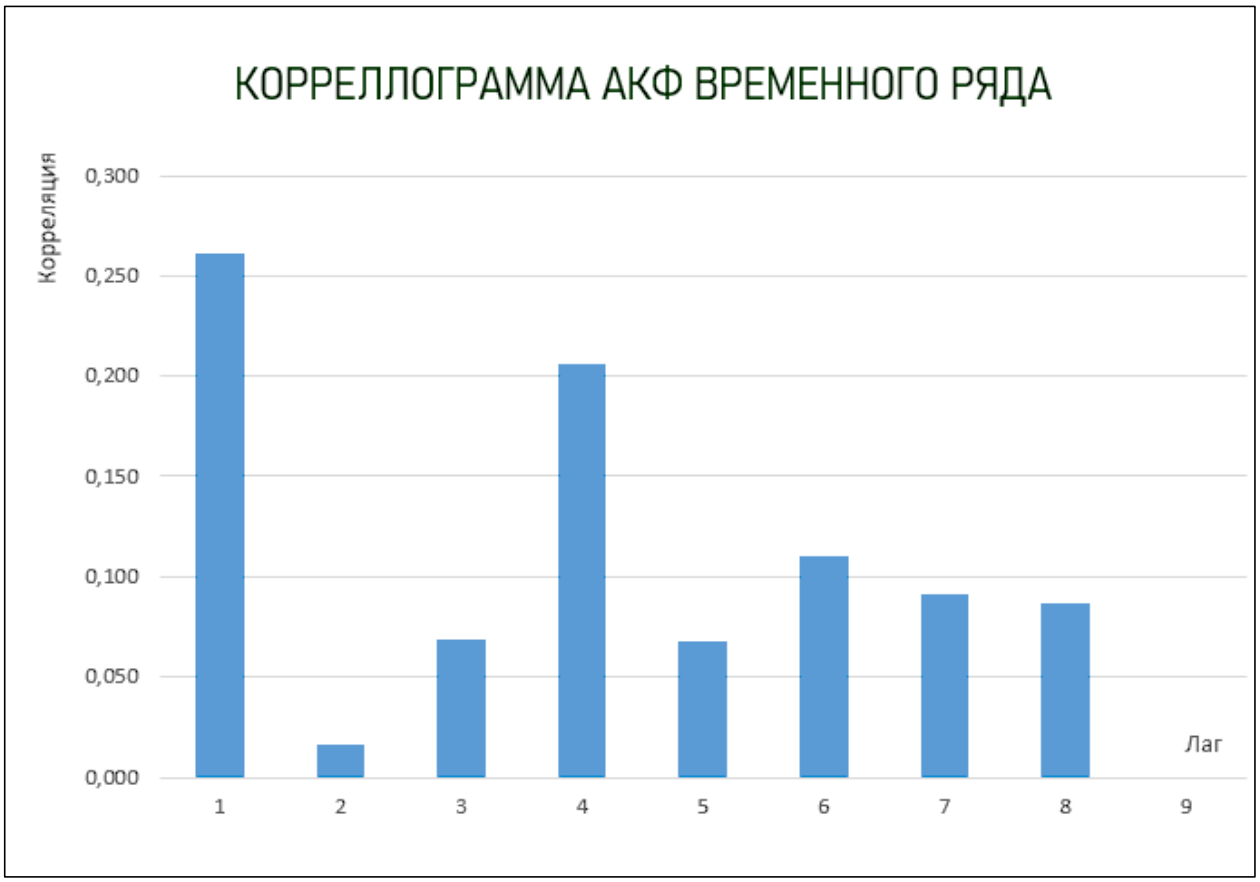


Рисунок 7 – Коррелограмма автокорреляции временного ряда

Полученным показателям тесноты связи необходимо дать качественную оценку. Для оценки силы автокорреляции переменной временного ряда используется шкала Чеддока, представленной в таблице 8.

Таблица 8. Шкала Чеддока для качественной оценки показателей тесноты связи

Количественная мера тесноты связи	Качественная характеристика силы связи
0,1 – 0,3	Слабая
0,3 – 0,5	Умеренная
0,5 – 0,7	Заметная
0,7 – 0,9	Высокая
0,9 – 0,99	Весьма высокая

В результате приведённых вычислений можно сделать вывод о том, что в данном временном ряду имеется крайне малая ниспадающая тенденция. В

динамике объемов производства наблюдаются периодические колебания, соответствующие зимнему сезону, затрудняющему добычу угля. На рисунке 8 представлена линия тренда для соответствующего временного ряда.

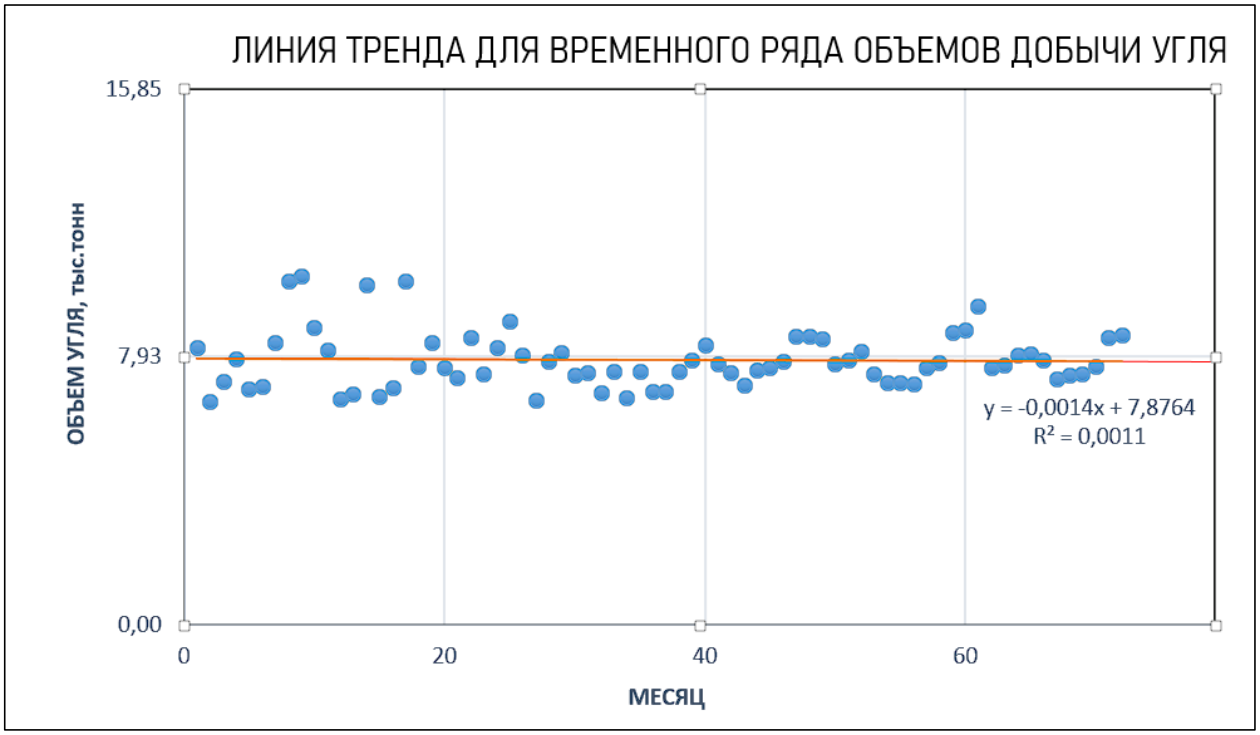


Рисунок 8 – Линия тренда временного ряда добычи угля в период с 2018 по 2023 год

Стоит отметить, что угольное направление сырьевого сектора промышленного производства Российской Федерации является экономикообразующим фактором для Государства. Следовательно, государственным аппаратом РФ принимаются дополнительные сторонние меры для обеспечения стабильности ценообразования на угольное сырье. Поэтому предсказуемо отсутствие резкого восходящего или нисходящего трендов. Сезонность же вызвана географическими условиями расположения ключевых угольных бассейнов и не связана с политикой и условиями добычи компании АО «СУЭК».

2.4.2 Статистический анализ функционалом Python

Производится определение набора данных, как переменной, в среде Google Collab для реализации программных решений, направленных на получение прогноза. Для получения примитивной статистической сводки по

данным используется метод *describe()* библиотеки Pandas. Данный метод предоставляет сжатую, краткую статистическую сводку по обрабатываемому набору данных. Результат выводится автоматически, без использования функции *print()* в образе таблицы. С помощью полученной таблицы можно наглядно отследить граничные значения квартилей набора данных, среднее значение по всей выборке, минимальную и максимальную границу выборки. Статистическая сводка по датасету представлена на рисунке 9.

mean	7.884444444444445
std	0.846589777711117
min	6.1
25%	7.2775
50%	7.735
75%	8.41
max	10.57

Рисунок 9 – Статистическая информация по датасету

Многие методы и модели основаны на предположениях о стационарности ряда. Реализуем тестирование Дикки-Фуллера на наличие единичных корней, чтобы определить стационарен ли исследуемый временной ряд. Результат проведения теста представлен на рисунке 10.

```
adf:  -10.403766132787482
p-value:  1.8798623951388763e-18
единичных корней нет, ряд стационарен
```

Рисунок 10 – Результаты теста Дикки-Фуллера на определение стационарности временного ряда

По результатам теста, результаты которого представлены на рисунке 8 – ряд стационарен, следовательно, его дисперсия, математическое ожидание и ковариация не будут изменяться со временем [6].

На этом этапе данные временных рядов должны быть разделены на X_train и y_train из обучающего набора, и X_test и y_test из тестового набора. После этого можно переходить к реализации моделей, так как сформированы и стандартизованы все необходимые виды выборок. Процесс

импорта в среду разработки Google Collab и стандартизации данных, формирование обучающих и тестовых выборок, импорт необходимых для работы библиотек представлен в листинге приложения А.

3 Проектирование информационной системы

3.1 Рекуррентные нейронные сети

Рекуррентные нейронные сети – класс нейронных сетей, ориентированных на анализ временной последовательности состояний группы объектов. В соответствии с приведенной характеристикой обозначим область применения рекуррентных нейронных сетей;

1) Статистический анализ данных;
2) Кластеризация и классификация данных;
3) Прогнозирование временных рядов.

Рассмотрим преимущества и недостатки рекуррентных нейронных сетей.

Преимущества:

- Обработка последовательных данных: рекуррентные НС отлично подходят для обработки данных, которые имеют временную зависимость, например, текст, речь, музыка, временные ряды. Они могут "помнить" предыдущую информацию, используя скрытые состояния, позволяющие им учитывать контекст и зависимость от предыдущих элементов последовательности.
- Обработка последовательностей переменной длины: рекуррентные НС не требуют фиксированной длины входных данных, как некоторые другие модели. Данная особенность делает модели этого типа наиболее подходящими для работы с последовательностями различной длины, например, предложениями в тексте или временными рядами разной продолжительности.
- Учет контекста: Благодаря способности "помнить" предыдущие элементы последовательности, НС этого класса могут использовать контекст для более точного предсказания или классификации.
- Изучение зависимостей: рекуррентные нейронные сети могут выявлять долгосрочные зависимости в данных, что позволяет им работать с

более сложными и многогранными задачами, например, машинным переводом, анализом текста, прогнозированием временных рядов.

Однако рекуррентные нейронные сети обладают и недостатками:

- Проблемы с долгосрочными зависимостями: НС данного класса могут испытывать трудности с запоминанием информации на протяжении длительных последовательностей, что приводит к "исчезновению градиента".
- Требовательность к объему поступающих данных: для проведения качественного обучения моделей этого класса необходимы большие датасеты, что в свою очередь увеличивает время обучения модели и увеличивает вероятность возникновения ошибок обучения, а при неправильно подобранных параметрах самой модели велик риск переобучения.

Рекуррентные нейронные сети состоят из входного слоя, скрытого слоя с функционалом памяти и выходного слоя. Базовая модель рекуррентной нейронной сети представлена на рисунке 11.

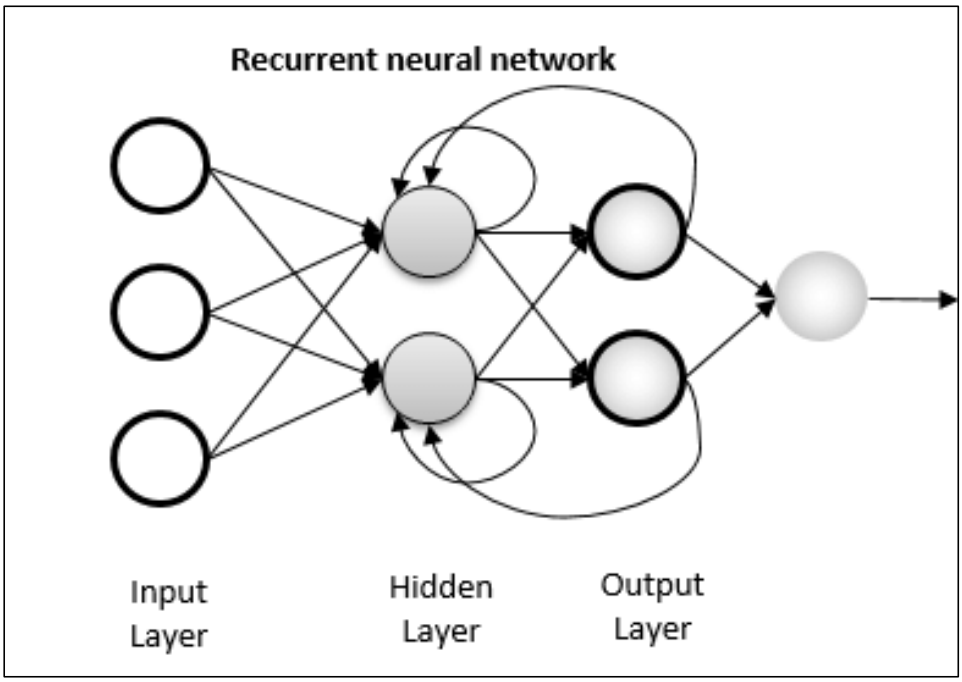


Рисунок 11 – Базовая архитектура рекуррентной нейронной сети

3.2 Проектирование RNN-сети для решения задачи прогнозирования

Рекуррентные нейронные сети (RNN) — это класс нейронных сетей, которые хороши для моделирования последовательных данных, таких как временные ряды или естественный язык.

Для сравнения результатов была реализована рекуррентная нейронная сеть (RNN) с четырьмя скрытыми слоями и плотным выходным слоем. Он использует функцию активации гиперболического тангенса *tanh*. Схема RNN-сети представлена на рисунке 12.

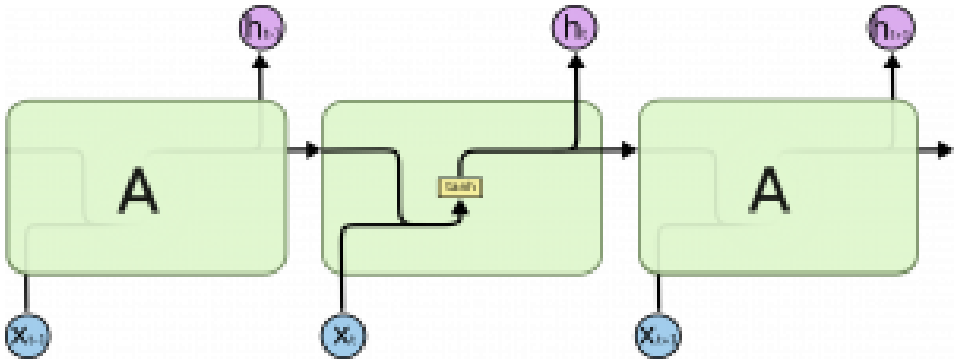


Рисунок 12 – Схема работы RNN-сети

Приведем описание и пояснение к выбору гиперпараметров строения модели RNN.

Количество слоев: меньшее количество предсказывает общий тренд, но не учитывает сезонные и прочие изменения.

Количество нейронов: подобрано с учетом оптимального соотношения времени обучения и объема модели.

Функция активации на выходном нейроне: не смотря на то, что данные нормализованны от 0 до 1, предсказанные данные могут быть как больше 1, так и меньше 0, следуя тренду данных, поэтому используется функция допускающая подобные значения. Данный вид нормализации значений называется минимаксной нормализацией, которая реализуется по формуле:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Данную формулу можно уточнить, приводя исходный набор значений к диапазону $[a, b]$:

$$X' = a + \frac{X - X_{min}}{X_{max} - X_{min}}(b - a)$$

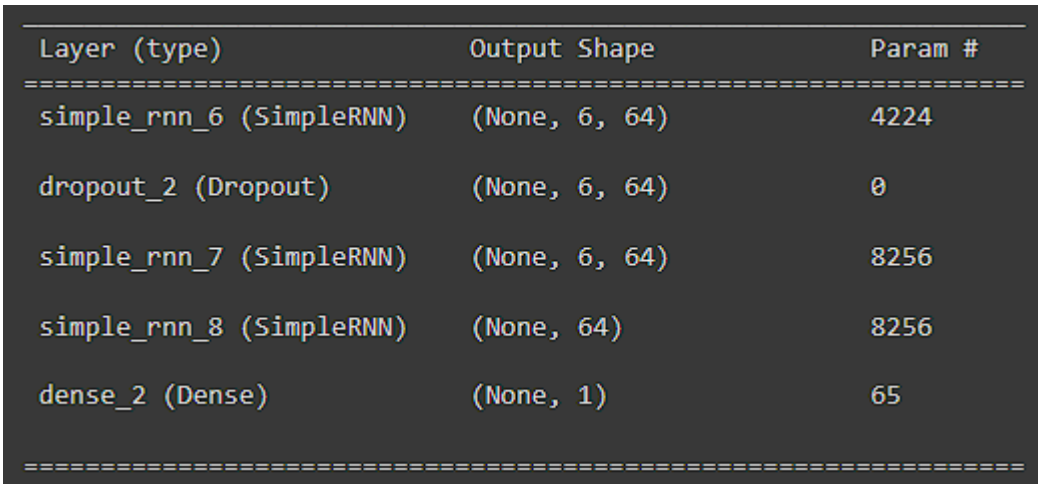
Наиболее распространенным методом минимаксной нормализации является приведение значений к диапазонам [0,1] и [-1,1].

Оптимизатор и функция ошибки: используются стандартные для задач предсказания временных рядов.

Параметры компиляции: размер *batch* выбран, исходя из скорости обучения модели, количество эпох методом проб, был выбран момент после которого скорость обучения модели снижается и модель постепенно перестает обучаться и начинает переобучаться.

Чтобы избежать переобучения, вводится dropout-слой исключения части нейронов с коэффициентом 0,2. компиляции и обучения модели: при большем размере batch_size модель будет. Для заданного объема данных количество эпох обучения равно 50-ти. Данное количество подбиралось экспериментальным путем до тех пор, пока качество предсказания по тестовой выборке не стало приемлемым. Большее количество эпох обучения приводит к переобучению модели, что отражалось как дестабилизация кривой с предсказанными значениями. Переобучение для данной модели происходит весьма быстро из-за ее упрощенного строения в сравнении с LSTM и GRU моделями.

Параметр batch_size равен 16 элементам. Изменение параметра batch_size влияет на скорость обучаться быстрее. Послойное строение реализованной RNN-модели представлено на рисунке 13 [8].



Layer (type)	Output Shape	Param #
simple_rnn_6 (SimpleRNN)	(None, 6, 64)	4224
dropout_2 (Dropout)	(None, 6, 64)	0
simple_rnn_7 (SimpleRNN)	(None, 6, 64)	8256
simple_rnn_8 (SimpleRNN)	(None, 64)	8256
dense_2 (Dense)	(None, 1)	65

Рисунок 13 – Послойная структура RNN-модели

Из рисунка 13 следует, что модель состоит из пяти слоев, а именно:

1) Входящий слой RNN на 64 нейрона и установленными входными данными длины равной шесть.

2) Дропаут (*dropout*) – слой, состоящий из 64-х нейронов, являющийся методом регуляризации в НС для предотвращения переобучения. Слой случайным образом отключает нейрон с вероятностью $p = 0{,}2$. Отключенным может быть любое множество нейронов входного и скрытых слоев.

3) Скрытый RNN слой на 64 нейрона – слой полносвязной RNN предлагаемый встроенными возможностями библиотеки Keras/Tensorflow.

4) Скрытый RNN слой на 64 нейрона, дублирующий структуру слоя №3.

5) Выходной слой с одним нейроном.

Реализация алгоритма описанной модели RNN – сети представлена в приложении Б.

Дополнительным параметром оценки качества работы модели служит время ее компиляции и получения результатов. Обозначим время компиляции RNN – модели нейронной сети в секундах как T_1. Для вышеперечисленных гиперпараметров модели и исследуемых данных $T_1 = 12{,}15$ секунд.

Результат обучения модели может быть представлен в графическом виде, путем отображения динамики параметров loss, отвечающего за

пропущенные значения. На рисунке 14 представлен график изменения характеристики модели loss в процессе обучения.

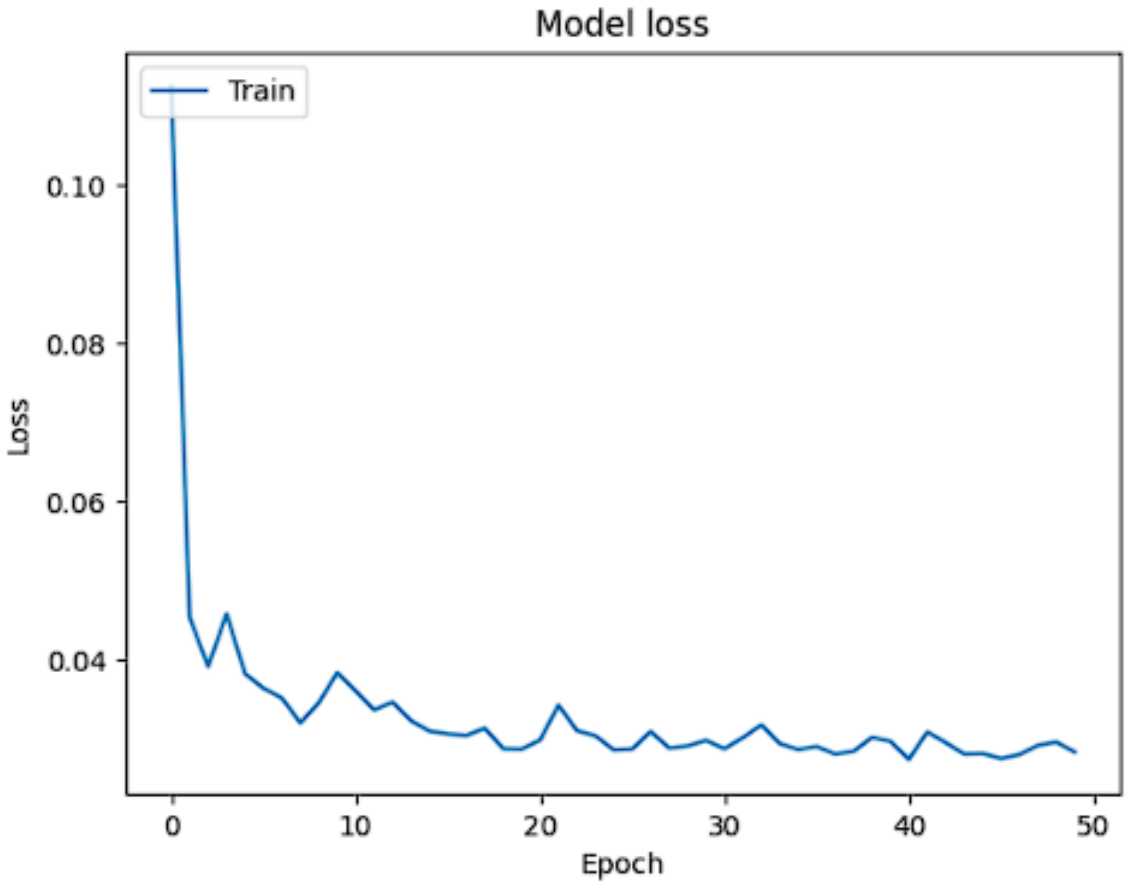


Рисунок 14 – Изменение ошибки обучения по эпохам для RNN – модели

По результатам, представленным на графике рисунка 14 – уточнить номер рисунка можно сделать вывод о том, что ошибка обучения на старте компиляции модели является весьма высокой, стремительно и быстро снижается, допуская малые колебания и в течение обучения стремится к минимуму. Такую динамику и размер ошибки обучения можно считать приемлемым. Следовательно, модель удовлетворяет требованиям, необходимым для решения поставленной задачи прогнозирования временного ряда.

Проведем тестирование RNN – модели нейронной сети. Тестирование обученной модели проводится на 15 % исходного набора данных, что составляет 72 значения. Требуется, чтобы результат тестирования в большей степени соответствовал динамике тестовой выборки, а именно перенимал

наличие тенденций, цикличностей и общий смысл, содержащийся в данных предложенных модели для тестирования.

На рисунке 15 представлен график результата тестирования RNN – модели нейронной сети. Кривой графика синего цвета соответствуют значения тестового набора данных. Оранжевая кривая – значения, предсказанные моделью на основе тестовых данных. Зеленая кривая – результат прогнозирования модели динамики изменения временного ряда на искомый период, т.е. помесячное предсказание на 2024 год.

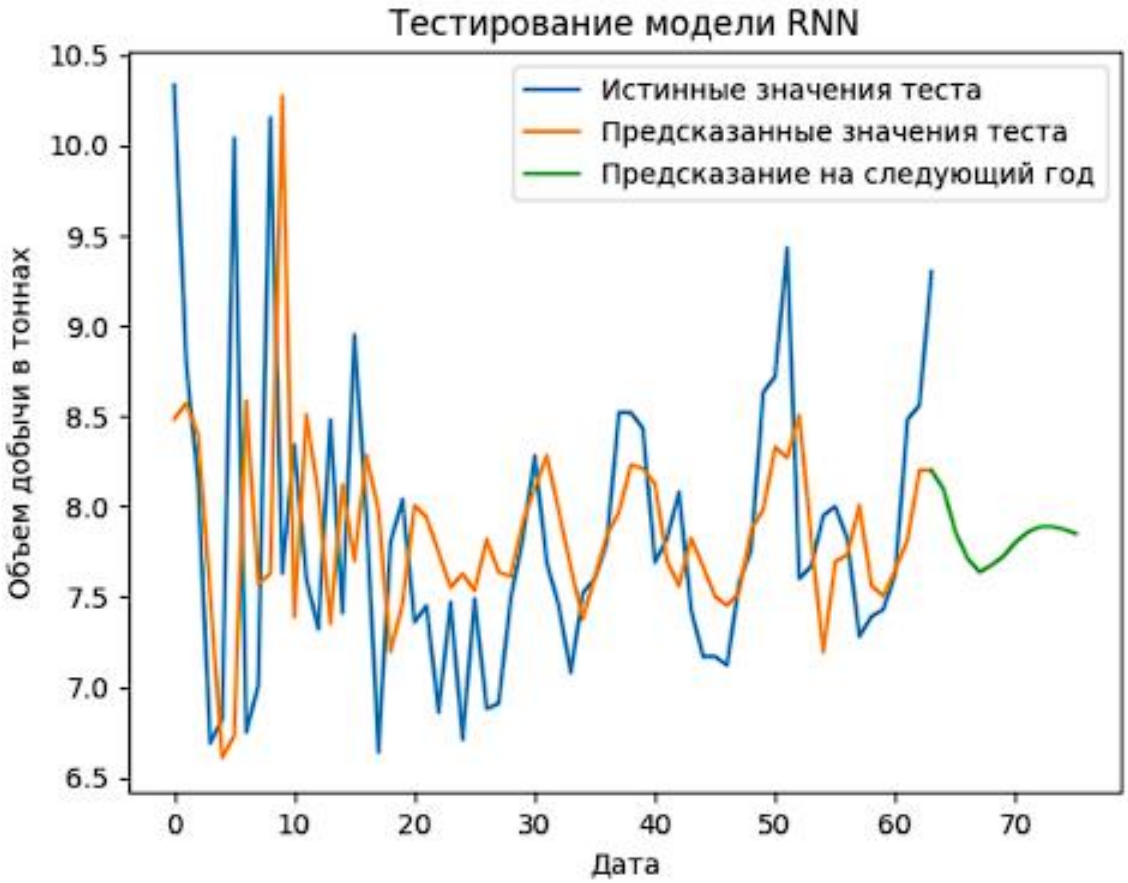


Рисунок 15 – Результат тестирования RNN – модели

3.3 Проектирование LSTM-сети для решения задачи прогнозирования

LSTM – рекуррентная нейронная сеть с долгой кратковременной памятью. С ее помощью есть возможность выявлять признаки из временных последовательностей, а также обрабатывать многомерные данные. При

обучении данная модель способна схватывать существенные детали прошлого контекста и сохранять их, пока они актуальны. В базовой версии LSTM состоит из ячеек. Все рекуррентные нейронные сети имеют форму цепочки повторяющихся модулей нейронной сети. В стандартных РНС этот повторяющийся модуль имеет простую структуру, например, один слой *tanh*. Схема модели представлена на рисунке 16.

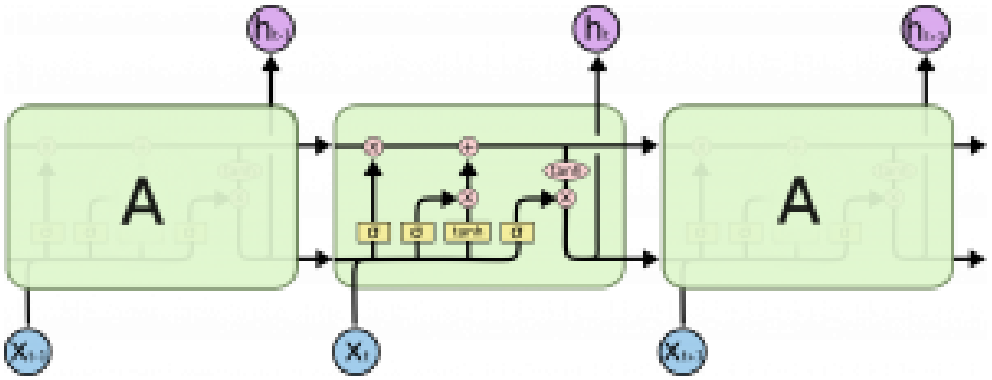


Рисунок 16 – Схема работы LSTM модели

На приведенной выше диаграмме каждая линия является вектором. Розовый круг означает поточечные операции, например, суммирование векторов. Под желтыми ячейками понимаются слои нейронной сети.

На схеме, представленной на рисунке 16, сигмовидный слой модели, обозначенный как σ, пропускает или не пропускает информацию. Состоит из операции поточечного умножения, в результате чего на выходе сигмовидного слоя выдаются числа от нуля до единицы определяя, сколько процентов каждой единицы информации пропустить дальше. Значение «0» означает «не пропустить ничего», значение «1» – «пропустить все».

Схема работы модели состоит из трёх этапов: слой утраты, слой сохранения и новое состояние. Для LSTM модели НС необходимо решить, какую информацию мы собираемся выбросить из состояния ячейки. Это решение принимается сигмовидным слоем, называемым «слоем утраты». Он получает на вход h и x и выдает число от 0 до 1 для каждого номера в состоянии ячейки C. Единица означает «полностью сохранить», а ноль — «полностью удалить» [7].

Приведем описание и пояснение к выбору гиперпараметров строения модели LSTM.

Количество слоев: меньшее количество предсказывает общий тренд, но не учитывает сезонные и прочие изменения, так же был добавлен дополнительный простой слой чтобы ускорить процесс выделения признаков из данных с LSTM слоев.

Количество нейронов: подобрано с учетом оптимального соотношения времени обучения и объема модели.

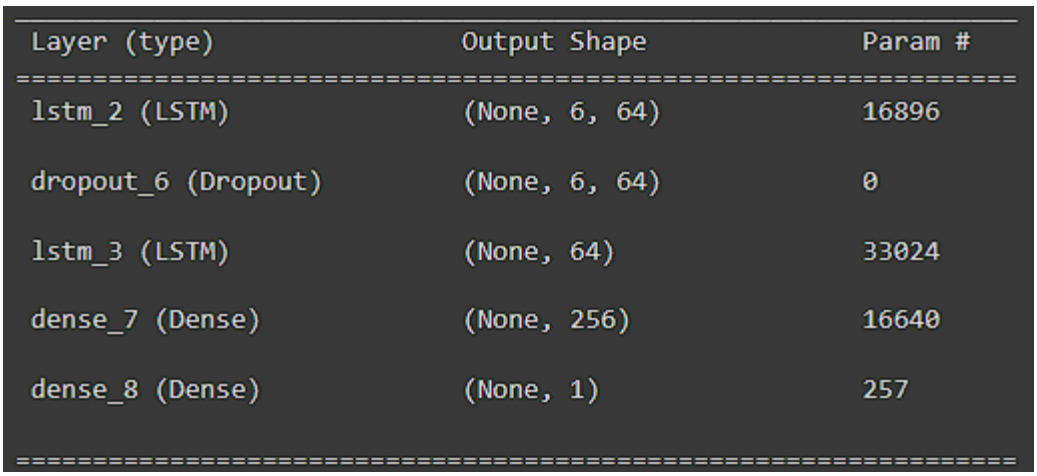
Функция активации на выходном нейроне: не смотря на то, что данные нормализованны от 0 до 1, предсказанные данные могут быть как больше 1, так и меньше 0, следуя тренду данных, поэтому используется функция допускающая подобные значения. Процесс и метод нормализации описан в пункте 3.2.

Оптимизатор и функция ошибки: используются стандартные для задач предсказания временных рядов.

Параметры компиляции: размер *batch* выбран, исходя из скорости обучения модели, количество эпох методом проб, был выбран момент после которого скорость обучения модели снижается и модель постепенно перестает обучаться и начинает переобучаться.

Чтобы избежать переобучения, вводится dropout-слой исключения части нейронов с коэффициентом 0,2. Для заданного объема данных количество эпох обучения равно 200. Изменение количества эпох обучения модели целиком влияет на то, какой результат покажет модель на этапе тестирования. Для данной модели наблюдалась весьма низкая скорость обучения, вследствие ее способности «запоминать» предшествующие признаки временных последовательностей. Поэтому для соблюдения баланса качества/скорости компиляции было подобрано количество эпох обучения в размере 200. Параметр batch_size равен 16 элементам. Изменение параметра batch_size влияет на скорость компиляции и обучения модели: при большем размере

batch_size модель будет обучаться быстрее. Послойное строение реализованной LSTM-модели представлено на рисунке 17.



Layer (type)	Output Shape	Param #
lstm_2 (LSTM)	(None, 6, 64)	16896
dropout_6 (Dropout)	(None, 6, 64)	0
lstm_3 (LSTM)	(None, 64)	33024
dense_7 (Dense)	(None, 256)	16640
dense_8 (Dense)	(None, 1)	257

Рисунок 17 – Послойная структура LSTM – модели

Из рисунка 17 следует, что модель состоит из пяти слоев, а именно:

1) Входящий слой LSTM типа на 64 нейрона и установленными входными данными длины равной шести.
2) Дропаут (*dropout*) – слой, состоящий из 64-х нейронов, являющийся методом регуляризации в НС для предотвращения переобучения. Слой случайным образом отключает нейрон с вероятностью p = 0,2. Отключенным может быть любое множество нейронов входного и скрытых слоев.
3) Скрытый LSTM слой на 64 нейрона – слой, предлагаемый встроенными возможностями библиотеки Keras/Tensorflow.
4) Скрытый линейный слой на 256 нейронов для ускорения обучаемости модели. Для данной модели задано кол-во эпох 200. Время компиляции необходимо было уменьшать, поэтому ввели данный слой.
5) Выходной слой с одним нейроном.

Реализация алгоритма описанной модели LSTM – сети представлена в приложении В.

Дополнительным параметром оценки качества работы модели служит время ее компиляции и получения результатов. Обозначим время компиляции LSTM – модели нейронной сети в секундах как T_2. Для вышеперечисленных гиперпараметров модели и исследуемых данных $T_2 = 53{,}25$ секунды.

Результат обучения модели может быть представлен в графическом виде, путем отображения динамики параметров loss, отвечающего за пропущенные значения. На рисунке 18 представлен график изменения характеристики модели loss в процессе обучения.

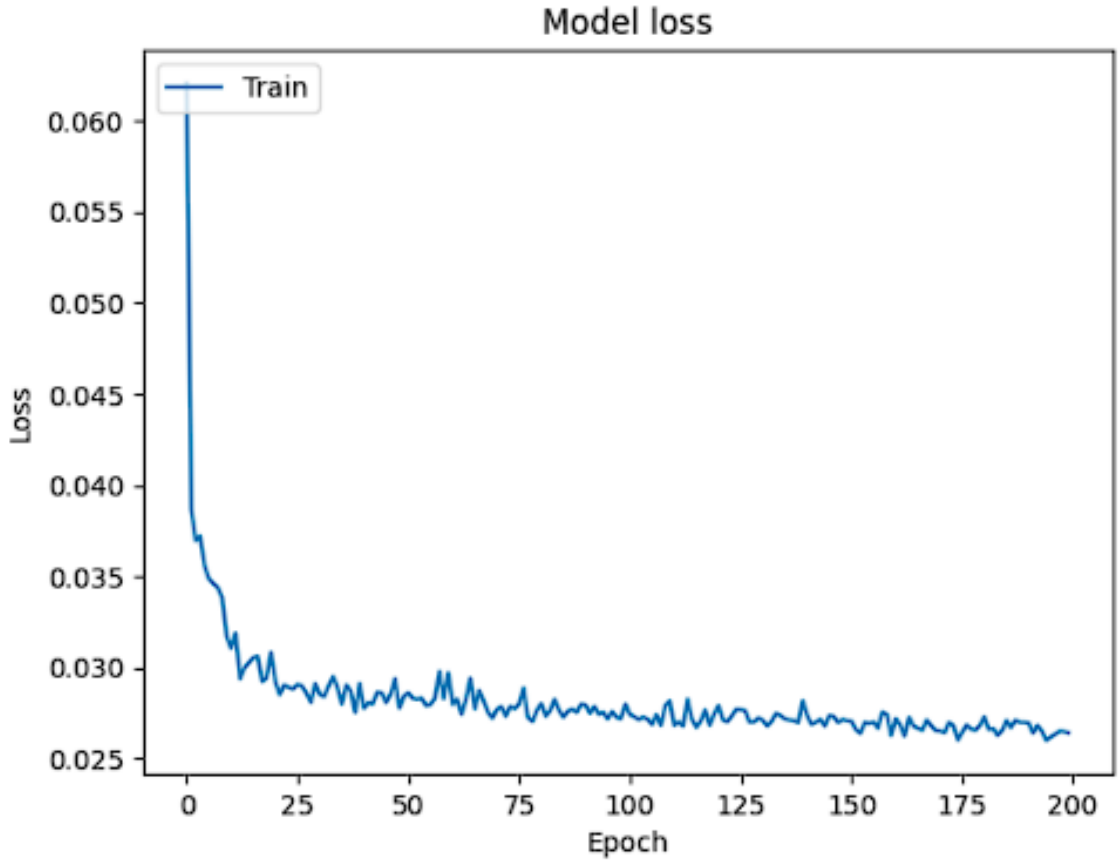


Рисунок 18 – Изменение ошибки обучения по эпохам для LSTM – модели

Из результатов, представленных на графике рисунка 18 можно сделать вывод о том, что ошибка обучения на старте компиляции модели является умеренно высокой, что ниже стартовой ошибки RNN – модели, стремительно и быстро снижается, допуская меньшие колебания и в течение обучения стремится к минимуму. Такую динамику и размер ошибки обучения можно считать приемлемым. Следовательно, модель удовлетворяет требованиям,

необходимым для решения поставленной задачи прогнозирования временного ряда.

Проведем тестирование LSTM – модели нейронной сети. Тестирование обученной модели проводится на 15 % исходного набора данных, а именно 72-х значениях от общего количества данных. Требуется, чтобы результат тестирования в большей степени соответствовал динамике тестовой выборки, а именно перенимал наличие тенденций, цикличностей и общий смысл, содержащийся в данных предложенных модели для тестирования.

На рисунке 19 представлен график результата тестирования RNN – модели нейронной сети. Кривой графика синего цвета соответствуют значения тестового набора данных. Оранжевая кривая – значения, предсказанные моделью на основе тестовых данных. Зеленая кривая – результат прогнозирования модели динамики изменения временного ряда на искомый период, т.е. помесячное предсказание на 2024 год.

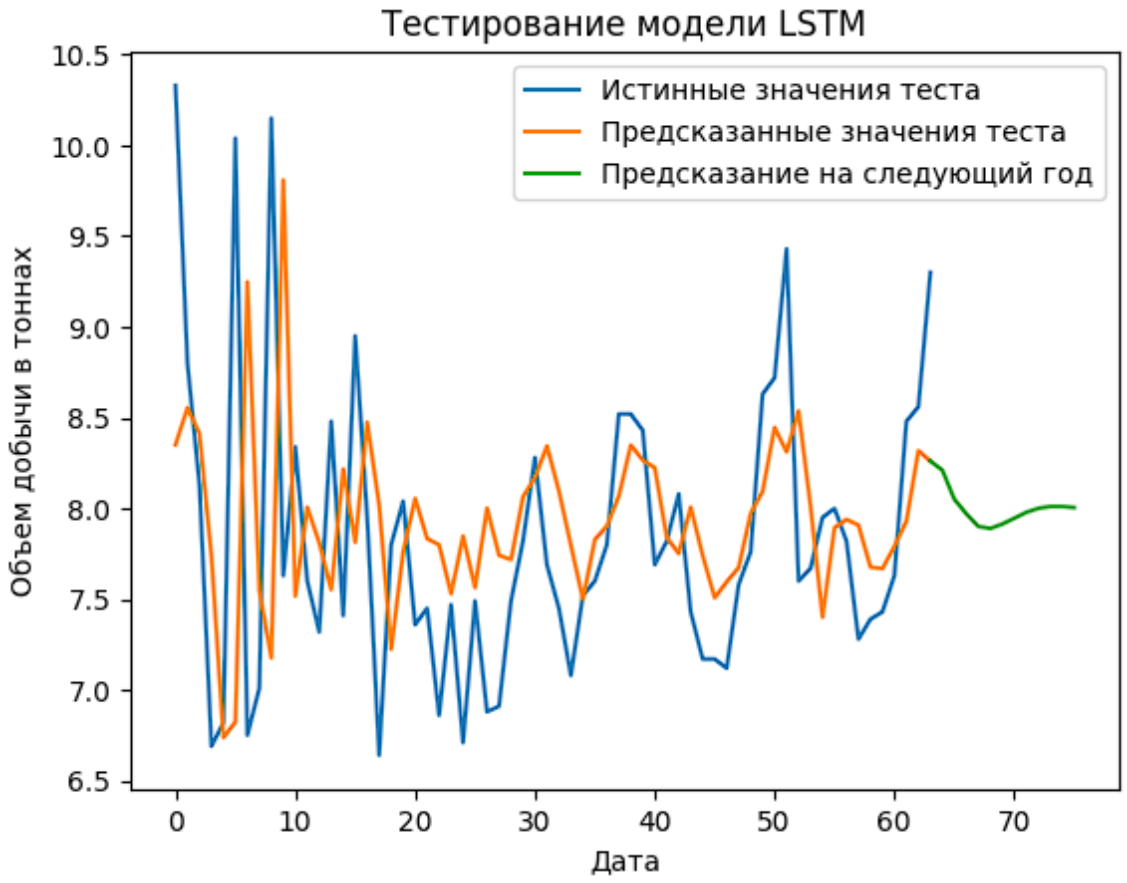


Рисунок 19 – Результат тестирования LSTM – модели

3.3 Проектирование GRU-сети для решения задачи прогнозирования

Модель нейронной сети GRU (Gated Recurrent Unit) – один из видов рекуррентных нейронных сетей, аналогичный LSTM-модели по качеству обучения и предоставления результата.

В отличие от LSTM, в GRU нет отдельного долгосрочного состояния ячейки. Состояние ячейки в GRU является комбинацией прошлого состояния и новых входных данных, модулируемых через обновляющие и сбрасывающие ворота. Это состояние обновляется на каждом шаге и переносит информацию по всей сети.

Выделим акцентные различия LSTM – модели и GRU – модели НС:

1) Состояние ячейки:
    - LSTM: включает отдельное состояние и скрытое состояние ячейки, что обеспечивает дополнительный контроль над информацией.
    - GRU: использует только скрытое состояние ячейки, что упрощает структуру модели, увеличивает скорость компиляции и обучения, однако приводит к незначительным искажениям полученных результатов.
2) Управление информацией:
    - LSTM: входные нейроны и нейроны «забывания» контролируют поток информации независимо друг от друга, что предрасполагает к большему контролю за регуляцией того, какую информацию необходимо сохранить, а какую следует отбросить.
    - GRU: ворота обновления контролируют информацию, сохраняемую из прошлого состояния и добавляет к ним новую информацию одновременно, что приводит к глобальной регуляции.

GRU модель отличается своей эффективностью при работе с простыми или маленькими наборами данных. Однако GRU модель плохо запоминает долгосрочные зависимости переменных.

Приведем описание и пояснение к выбору гиперпараметров строения модели GRU.

Количество слоев: меньшее количество предсказывает общий тренд, но не учитывает сезонные и прочие изменения, так как данной моделью используется только скрытое состояние ячейки.

Количество нейронов: подобрано с учетом оптимального соотношения времени обучения и объема модели.

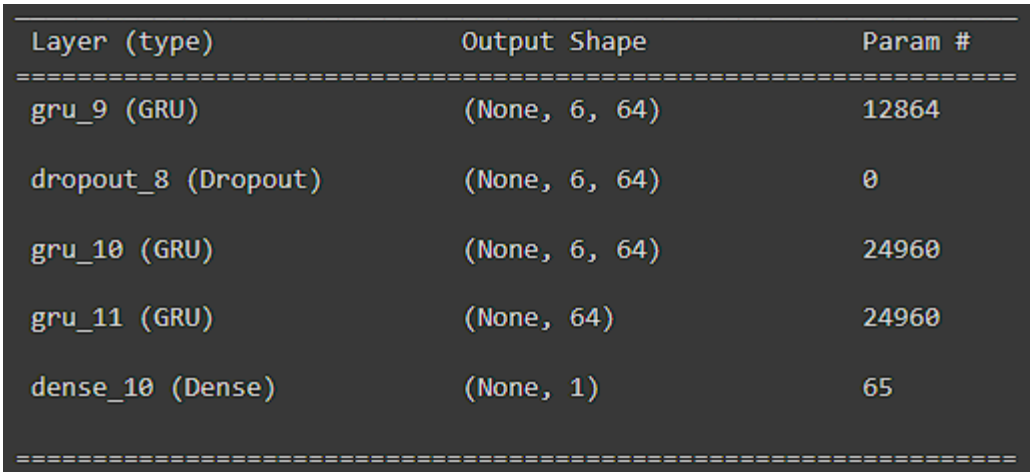
Функция активации на выходном нейроне: не смотря на то, что данные нормализованны от 0 до 1, предсказанные данные могут быть как больше 1, так и меньше 0, следуя тренду данных, поэтому используется функция допускающая подобные значения. Процесс и метод нормализации описан в пункте 3.2.

Оптимизатор и функция ошибки: используются стандартные для задач предсказания временных рядов.

Параметры компиляции: размер batch выбран, исходя из скорости обучения модели, количество эпох методом проб, был выбран момент, после которого скорость обучения модели снижается и модель постепенно перестает обучаться и начинает переобучаться.

Чтобы избежать переобучения, вводится dropout-слой исключения части нейронов с коэффициентом 0,2. Так как модель GRU сочетает в себе простоту RNN и качество управления информацией LSTM-модели, время, требующееся для ее обучения будет значительно меньшим чем для LSTM-модели. Однако, из-за отсутствия долгосрочного запоминания предшествующих закономерностей и особенностей данных, модель GRU рискует переобучаться на меньших количествах эпох, чем LSTM-модель. Экспериментальным путем отслеживания результатов тестировки модели и качества ее предсказания для заданного объема данных количество эпох обучения равно 100. Параметр

batch_size равен 16 элементам. Послойное строение реализованной GRU - модели представлено на рисунке 20.



Layer (type)	Output Shape	Param #
gru_9 (GRU)	(None, 6, 64)	12864
dropout_8 (Dropout)	(None, 6, 64)	0
gru_10 (GRU)	(None, 6, 64)	24960
gru_11 (GRU)	(None, 64)	24960
dense_10 (Dense)	(None, 1)	65

Рисунок 20 – Послойная структура GRU – модели

Из рисунка 13 следует, что модель состоит из пяти слоев, а именно:

1) Входящий слой GRU на 64 нейрона и установленными входными данными длины равной шесть.

2) Дропаут (dropout) – слой, состоящий из 64-х нейронов, являющийся методом регуляризации в НС для предотвращения переобучения. Слой случайным образом отключает нейрон с вероятностью p = 0,2. Отключенным может быть любое множество нейронов входного и скрытых слоев.

3) Скрытый GRU слой на 64 нейрона – слой, предлагаемый встроенными возможностями библиотеки Keras/Tensorflow.

4) Скрытый GRU слой на 64 нейрона, дублирующий структуру слоя №3.

5) Выходной слой с одним нейроном.

Реализация алгоритма описанной модели GRU – сети представлена в приложении Г.

Дополнительным параметром оценки качества работы модели служит время ее компиляции и получения результатов. Обозначим время компиляции GRU – модели нейронной сети в секундах как T_3. Для вышеперечисленных гиперпараметров модели и исследуемых данных T_3 = 37,98 секунды.

Результат обучения модели может быть представлен в графическом виде, путем отображения динамики параметров loss, отвечающего за пропущенные значения. На рисунке 21 представлен график изменения характеристики модели loss в процессе обучения.

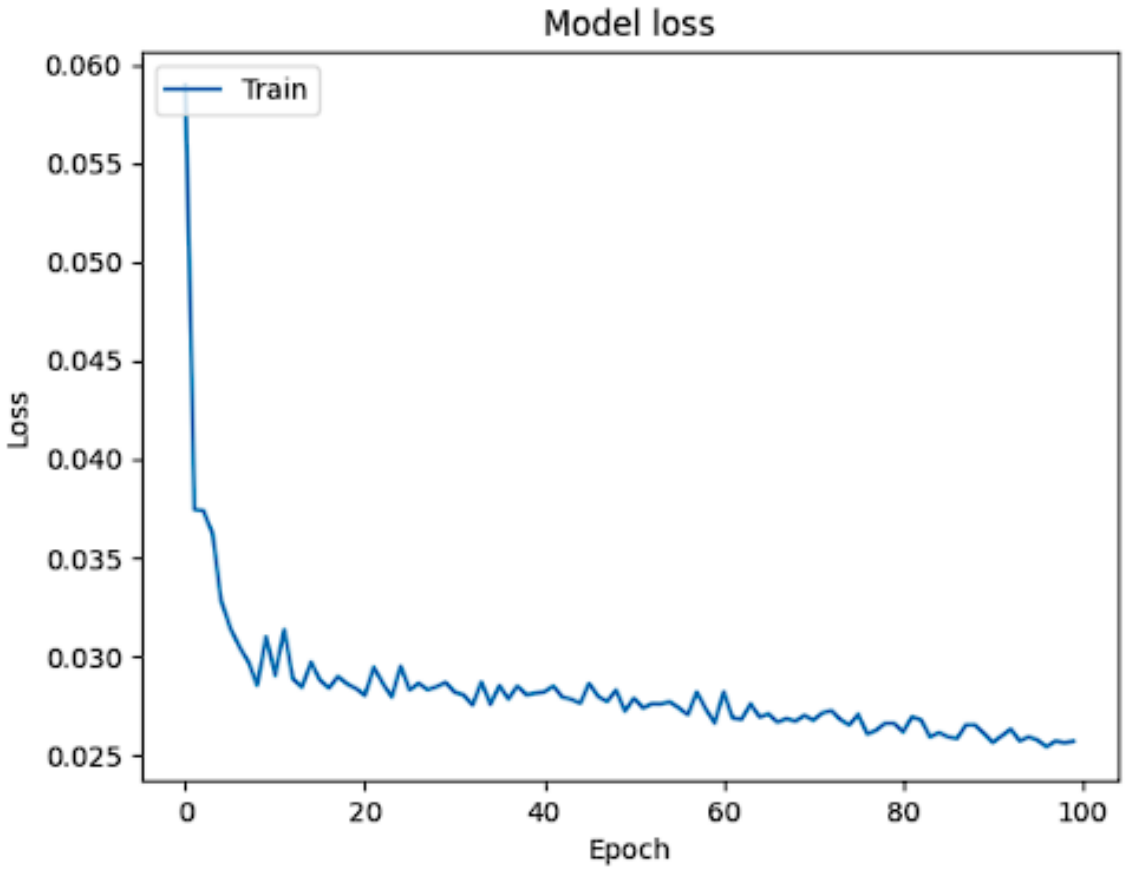


Рисунок 21 – Изменение ошибки обучения по эпохам для GRU – модели

Из результатов, представленных на графике рисунка 21 можно сделать вывод о том, что ошибка обучения на старте компиляции модели является наименьшей из представленных моделей RNN и LSTM. Ошибка обучения стремительно и быстро снижается, допуская меньшие колебания и в течение обучения стремится к минимуму. Такую динамику и размер ошибки обучения можно считать приемлемым. Следовательно, модель удовлетворяет требованиям, необходимым для решения поставленной задачи прогнозирования временного ряда.

Проведем тестирование GRU – модели нейронной сети. Тестирование обученной модели проводится на 15 % исходного набора данных. Требуется,

чтобы результат тестирования в большей степени соответствовал динамике тестовой выборки, а именно перенимал наличие тенденций, цикличностей и общий смысл, содержащийся в данных предложенных модели для тестирования.

На рисунке 22 представлен график результата тестирования GRU – модели нейронной сети. Синей кривой графика соответствуют значения тестового набора данных. Оранжевая кривая – значения, предсказанные моделью на основе тестовых данных. Зеленая кривая – результат прогнозирования модели динамики изменения временного ряда на искомый период, т.е. помесячное предсказание на 2024 год.

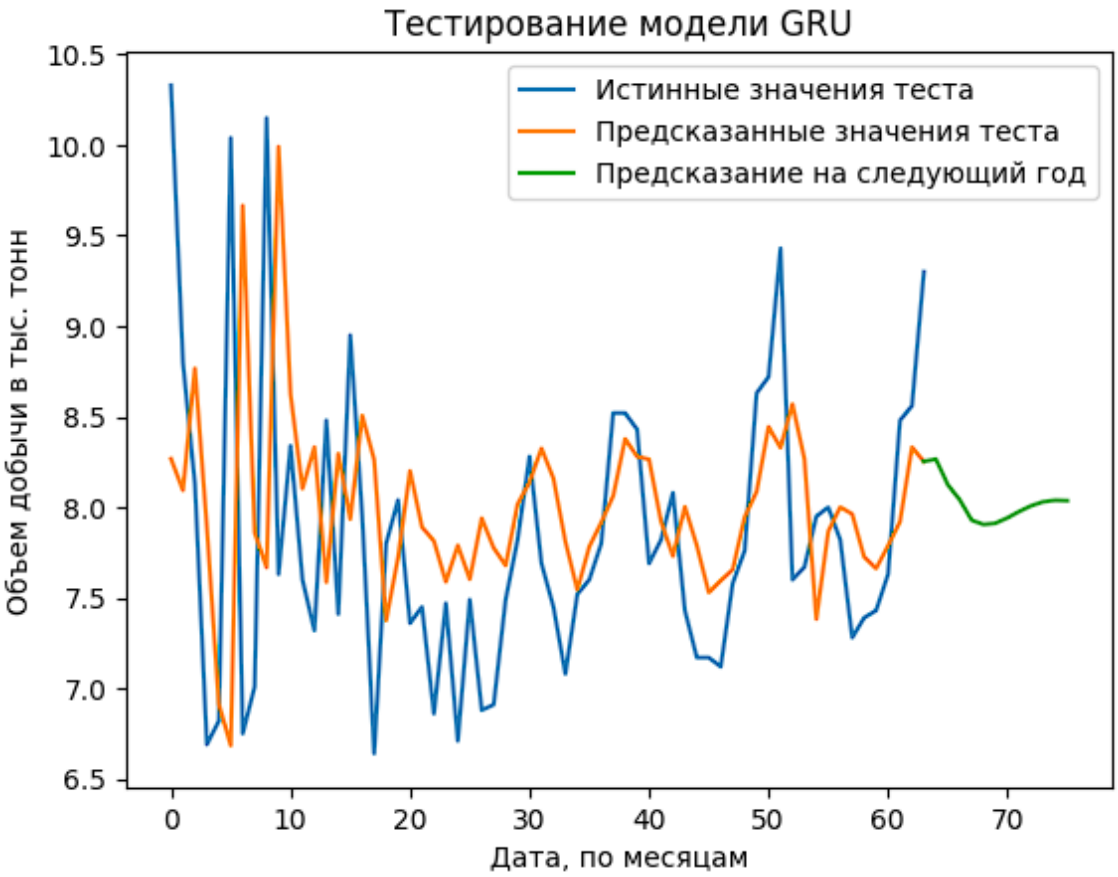


Рисунок 22 – Результат тестирования GRU - модели

4 Оценка результатов

4.1 Демонстрация результатов работы моделей прогнозирования

По результатам обучения моделей RNN, LSTM и GRU нейронных сетей было реализовано построение сводного графика результатов прогнозирования каждой из модели. Сравнение результата работы моделей представлено на рисунке 23.

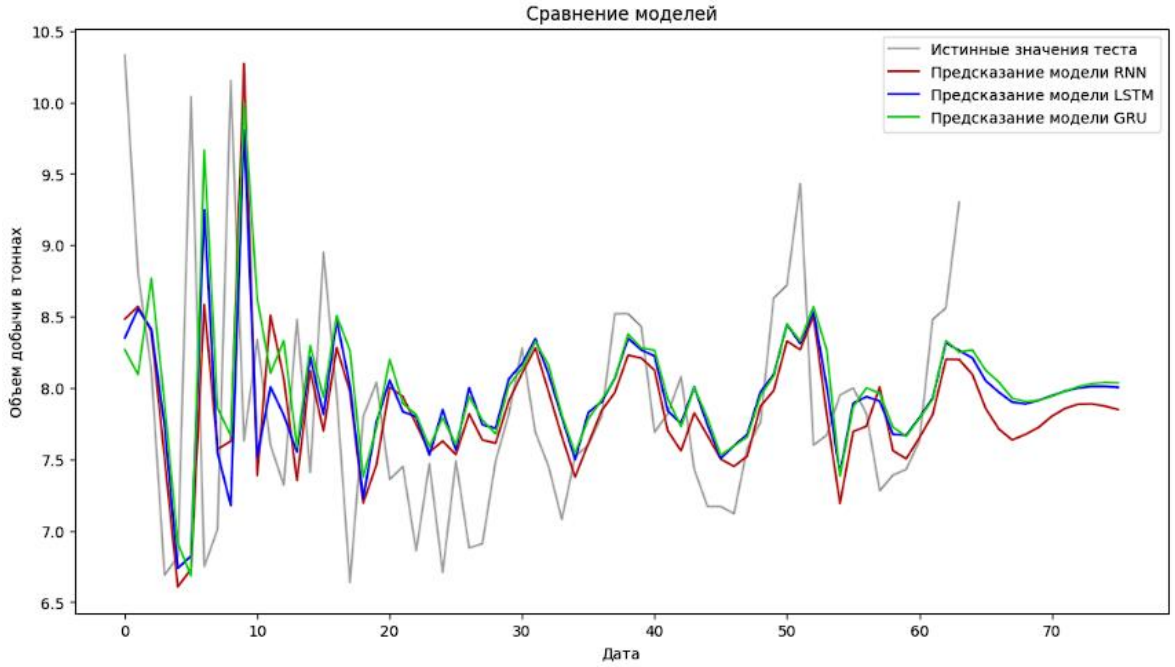


Рисунок 23 – Результат прогнозирования моделей

По результатам, представленным на графике, сложно не отметить, что модель RNN справилась с прогнозированием на тестовой выборке лучше всего. Такое сравнение не говорит о том, что работа остальных моделей шла некорректно или результаты их прогноза неприемлемы. Такой результат свидетельствует лишь о том, что модель в большей степени справляется с отслеживанием и соблюдением тенденций и поведений переменной. Об этом может говорить более амплитудное движение кривой, в целях повторения исходной динамики (в особенности пиковых элементов графика).

Наблюдается схожесть в результатах прогнозирования LSTM и GRU моделей, что подтверждается их родственными характеристиками. Об этом свидетельствует практически полное наложение кривых графика, изображенного на рисунке 21, соответствующих линиям прогноза LSTM и

GRU моделей. Наблюдается сглаживание кривых, что говорит об усреднении предсказываемых моделями значений.

График, представленный на рисунке 23 дает понять, что все три реализованные модели корректно справляются с поставленной задачей прогнозирования временного ряда, содержащего данные помесячного производства угля АО «СУЭК». Соблюдены сезонная составляющая и тенденции. На полученных прогнозах также наблюдается спад производительности компании в зимний период (подразумевается диапазон месяцев ноябрь – март) и рост производительности в остальное время года. Данные характерности наблюдаются для исходного временного ряда, результаты тестирования моделей им соответствуют.

Рассмотрим «хвосты» линий предсказания моделей в увеличенном масштабе. На рисунке 24 представлен график прогнозирования объемов добычи угля моделями нейронных сетей на 2024 год по месяцам.

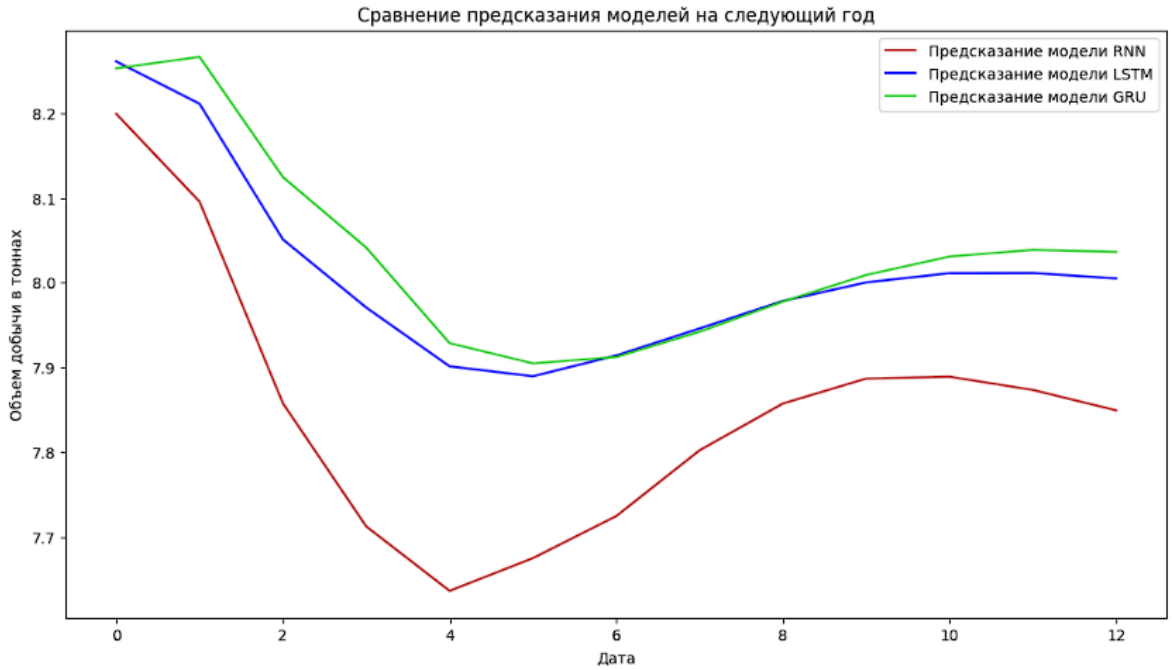


Рисунок 24 – Прогнозирование объемов добычи угля моделями НС на 2024 год

При рассмотрении графика, представленного на рисунке 24, наблюдаем сезонную составляющую, описанную выше как падение объемов добычи угля в холодное время года. По оси X графика отложены порядковые номера

месяцев 2024 года, а именно временной интервал от 01.01.2024 по 01.12.2024. Все три модели определили тенденцию корректно, продемонстрировав дальнейший рост объемов добычи с 4-го месяца, а именно с марта 2024 года, когда упрощаются условия добычи угля ввиду таяния снегов и размягчения горных пород вследствие повышения среднесуточной температуры.

Основываясь на полученной визуализации результатов (см. Рисунок 23-24) можно оценить качество предсказания как приемлемое. Данный результат удовлетворителен, но может улучшаться за счет изменения параметров моделей. Переобучения моделей допущено не было.

Алгоритм реализации вышеперечисленных графиков представлен в листинге Г.

Так как качество прогнозирования оценивается по значению ошибки обучения модели, проведем сравнение качества обучения на основе параметра loss, характеризующего степень отклонения значения, определенного моделью, от соответствующего ему значения выборки. Сводный график ошибок прогнозирования для трех моделей представлен на рисунке 25.

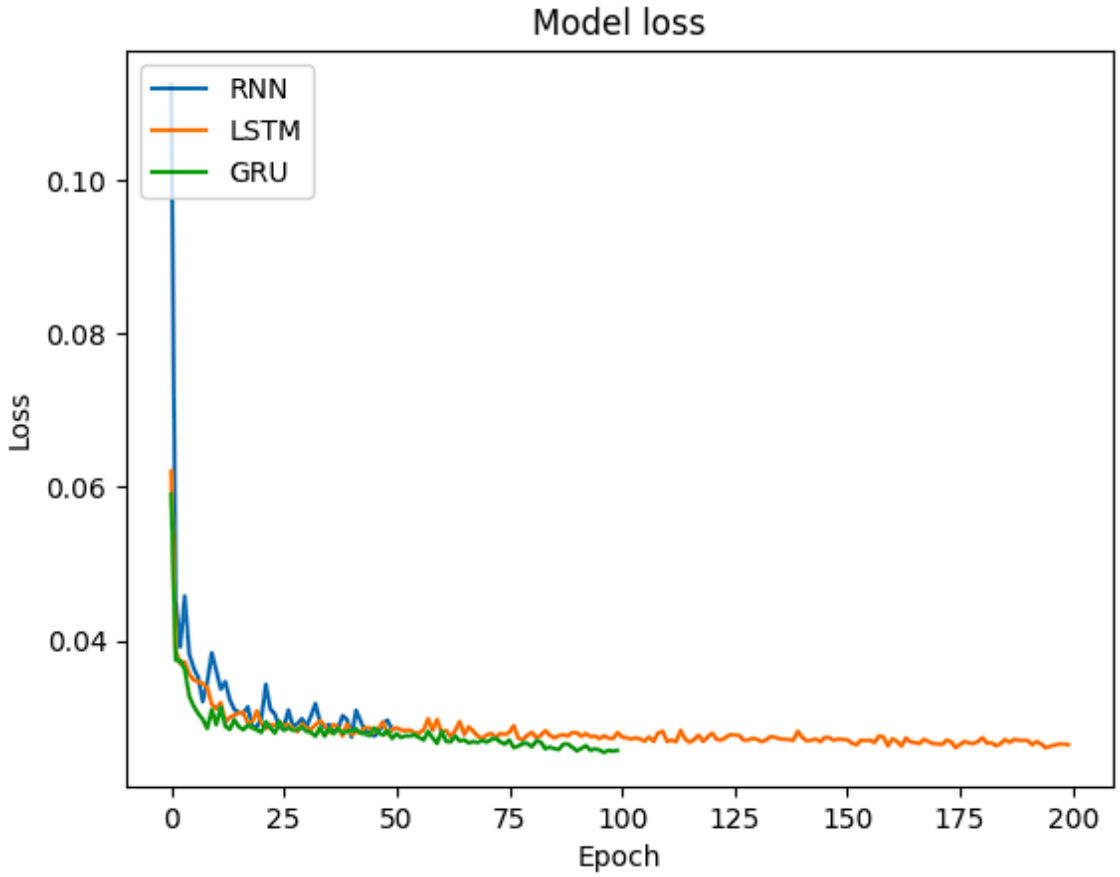


Рисунок 25 – Ошибка прогнозирования моделей НС RNN, LSTM, GRU

По результатам графика видно, что для RNN – модели наблюдается наибольшая стартовая ошибка обучения. GRU – модель показала наиболее быстрое снижение размера ошибки, а модель LSTM показала себя стабильной относительно скорости снижения ошибки обучения (скорость характеризуется количеством эпох, за которые ошибка достигла минимума) и качества обучения.

Конечная ошибка (ошибка модели на момент окончания обучения) для каждой модели следующая:

- Ошибка модели RNN: 0,02834
- Ошибка модели LSTM: 0,02642
- Ошибка модели GRU: 0,02563

Полученные показатели свидетельствуют о том, что каждая из описанных и реализованных моделей подходит для решения поставленной задачи, корректно обучается и обрабатывает предложенные ей данные.

Удовлетворительными для расчета прогнозируемой прибыли предприятия АО «СУЭК» с прогнозируемого объема добычи угля считаем результаты всех трех моделей.

Результат прогнозирования объема добычи угля в тыс. тонн каждой модели на 2024-й год по месяцам представлен в таблице 9.

Таблица 9. Результат прогнозирования объема добычи угля на 2024-й год по месяцам.

Номер месяца	Прогноз RNN-модели, тыс.тонн	Прогноз LSTM-модели, тыс.тонн	Прогноз GRU-модели, тыс.тонн
1	8.10	8.21	8.27
2	7.86	8.05	8.12
3	7.71	7.97	8.04
4	7.64	7.90	7.93
5	7.67	7.89	7.90
6	7.72	7.91	7.91
7	7.80	7.95	7.94
8	7.86	7.98	7.98
9	7.89	8.00	8.01
10	7.89	8.01	8.03
11	7.87	8.01	8.04
12	7.85	8.00	8.04

4.2 Расчет прогнозируемой прибыли промышленного предприятия

Так как стоимость одной тонны угля C_{2024} – годовой показатель, считаем, что в течение года стоимость угля оставалась неизменной. Тогда, чтобы рассчитать прибыль, полученную предприятием за 2024 год необходимо вычислить произведение прогнозируемой цены за одну тонну угля на значения прогноза объемов добычи угля по месяцам, представленным в таблице 9. Полученные значения прибыли представим в виде таблицы 10.

Таблица 10. Прибыль промышленного предприятия АО «СУЭК» от добычи угля на 2024-й год по месяцам.

Номер месяца	Прогнозная прибыль RNN-модели, тыс. р.	Прогнозная прибыль LSTM-модели, тыс. р.	Прогнозная прибыль GRU-модели, тыс. р.
1	27216,00	27585,60	27787,20
2	26409,60	27048,00	27283,20
3	25905,60	26779,20	27014,40
4	25670,40	26544,00	26644,80
5	25771,20	26510,40	26544,00
6	25939,20	26577,60	26577,60
7	26208,00	26712,00	26678,40
8	26409,60	26812,80	26812,80
9	26510,40	26880,00	26913,60
10	26510,40	26913,60	26980,80
11	26443,20	26913,60	27014,40
12	26376,00	26880,00	27014,40

Представим динамику изменения месячной прибыли АО «СУЭК» с продажи добытого угля на прогнозный период на рисунке 26.

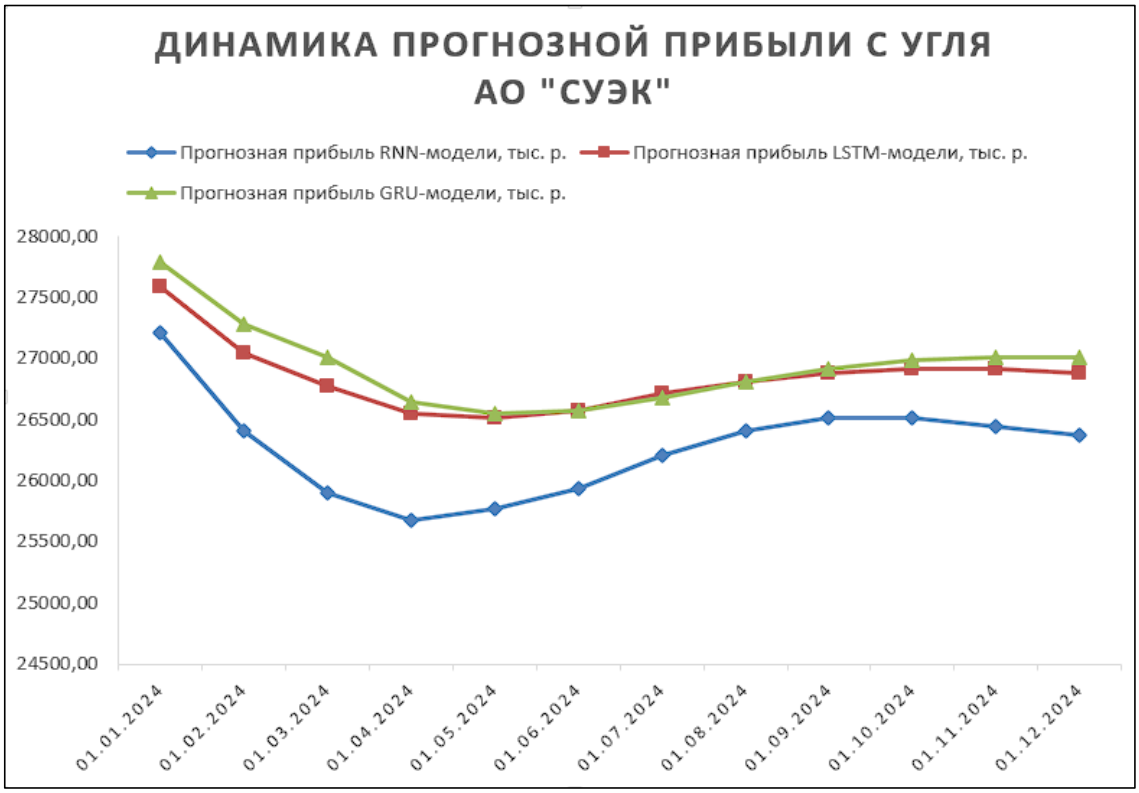


Рисунок 26 – Динамика прибыли АО «СУЭК» на прогнозный период

ЗАКЛЮЧЕНИЕ

В ходе проделанной работы, с использованием инструментов библиотек языка программирования Python, пакетом анализа данных MS Excel и функций статистической обработки и анализа временных рядов был реализован прогноз прибыли промышленного предприятия АО «СУЭК».

В процессе выполнения работы в рамках сформулированных задач было выполнено:

1. Поиск и сбор статистических данных о добыче угля компанией АО «СУЭК». Поиск и сбор статистических данных макроэкономических параметров, таких как темп инфляции и индекс инфляции Российской Федерации за период 2018 – 2023 гг.
2. Анализ и стандартизация полученных данных. Формирование обучающих и тренировочных выборок.
3. Прогнозирование стоимости единицы товара, а именно получение данных о стоимости одной тонны угля на прогнозный период.
4. Прогнозирование и выведение индекса инфляции на прогнозный период.
5. Реализация процесса построения и обучения моделей нейронных сетей рекуррентного класса, а именно RNN, LSTM, GRU моделей.
6. Реализация графического представления полученных результатов прогнозирования моделей.
7. Сравнительный анализ качества работы моделей, точности предоставления прогноза и прочих характеристиках моделей.
8. Расчет прогнозируемой прибыли промышленного предприятия АО «СУЭК» на основании полученных прогнозных данных о стоимости единицы товара и данных об объеме добытого угля, спрогнозированных нейронными сетями RNN, LSTM, GRU.

Таким образом, данной работе были рассмотрены основные аспекты задачи решения задачи прогнозирования показателей выработки угля сырьевым сектором российской Федерации. Были описаны этапы

формирования набора данных, методы их обработки и представления в машиночитаемый вид. Были рассмотрены модели нейронных сетей, использованные для решения данной задачи. Определен вектор совершенствования моделей в плане расширения функционала инструмента предсказания, а также оптимизация реализованных решений.

На основании балансов, отчетов о финансовом состоянии предприятия, а также годовых отчетов компании, был проведен анализ технико-экономических показателей АО «СУЭК», который показал, что в настоящий момент предприятие наращивает объемы производства, тем самым увеличивая выручку и повышая показатели рентабельности.

В рамках поставленной задачи необходимо отметить, что результаты прогнозирования каждой из моделей остались удовлетворительными. Следовательно, требуется сравнение параметров оптимальности и рациональности использования моделей для определения наилучшей.

Из пунктов главы 3 следует, что LSTM-модель обладает наиболее сложной структурой построения, способна качественнее остальных отследить, соблюсти и запоминать особенности поведения переменной временного ряда. Стоит подчеркнуть, что модель отлично подходит для построения прогнозов как на малых, так и на больших объемах данных, сводя потери к минимуму, лишь улучшая качество прогноза (при увеличении обучающих выборок и корректировки значений гиперпараметров под условия поставленной задачи).

RNN и GRU модели показали меньшее время компиляции, однако они предоставляют прогноз с большими погрешностями ввиду их неспособности запоминать длительные закономерности и тенденции последовательностей.

Таким образом, в рамках ВКР считается, что LSTM-модель нейронной сети показала наилучший результат в предоставлении прогноза. Данная модель может использоваться в прогнозировании показателей промышленных предприятий.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Кирчевская П. В., Яковлева Е. А. Исследование процессов ценообразования в IT-сфере // Обработка, передача и защита информации в компьютерных системах ’24: Четвертая Междунар. науч. конф. (СПб., 8 – 15 апреля 2024): сб. докл. – СПб.: ГУАП, 2024 – стр.101 – 104. (дата обращения: 08.04.2024)

2. Коврижных О.Е., Петрова О.М. Прогнозирование в экономических системах на основе нейронных сетей // Экономика и социум. 2014. №1-3 (10). URL: https://cyberleninka.ru/article/n/prognozirovanie-v-ekonomicheskih-sistemah-na-osnove-neyronnyh-setey (дата обращения: 10.04.2024).

3. Виноградова Светлана Сергеевна, Касимов Николай Николаевич Применение нейросетевых технологий с целью оптимизации управления судостроительным предприятием (на примере Астраханского региона) // Вестник АГТУ. Серия: Морская техника и технология. 2011. №2. URL: https://cyberleninka.ru/article/n/primenenie-neyrosetevyh-tehnologiy-s-tselyu-optimizatsii-upravleniya-sudostroitelnym-predpriyatiem-na-primere-astrahanskogo-regiona (дата обращения: 11.04.2024).

4. Ляшева С.А., Шлеймович М.П., Кирпичников А.П., Гришина О.Д. Нейросетевое прогнозирование фугасности индивидуальных взрывчатых веществ // Вестник Казанского технологического университета. 2015. №17. URL: https://cyberleninka.ru/article/n/neyrosetevoe-prognozirovanie-fugasnosti-individualnyh-vzryvchatyh-veschestv (дата обращения: 11.04.2024).

5. Гайнуллин Р.Н., Рахал Я., Ризаев И.С., Шарнин Л.М. Прогнозирование бизнес-процессов на основе нейронных сетей // Вестник Казанского технологического университета. 2017. №3. URL: https://cyberleninka.ru/article/n/prognozirovanie-biznes-protsessov-na-osnove-neyronnyh-setey (дата обращения: 11.04.2024).

6. Боев А.Г., Пузаков А.Г., Анисимов Ю.П. Оптимизация бюджета стратегии преобразований промышленного комплекса на основе

нейросетевого моделирования // Статистика и экономика. 2022. №3. URL: https://cyberleninka.ru/article/n/optimizatsiya-byudzheta-strategii-preobrazovaniy-promyshlennogo-kompleksa-na-osnove-neyrosetevogo-modelirovaniya (дата обращения: 11.04.2024).

7. Хрищатый А.С. Исследование использования нейросетей для анализа данных и принятия бизнес-решений: анализ эффективности использования нейросетей для обработки больших объемов данных и предоставления ценных инсайтов для принятия решений // Инновации и инвестиции. 2023. №7. URL: https://cyberleninka.ru/article/n/issledovanie-ispolzovaniya-neyrosetey-dlya-analiza-dannyh-i-prinyatiya-biznes-resheniy-analiz-effektivnosti-ispolzovaniya (дата обращения: 18.04.2024).

8. Хамхоева Ф.Я. Нейронные сети в экономическом анализе: плюсы и минусы // Norwegian Journal of Development of the International Science. 2020. №51-4. URL: https://cyberleninka.ru/article/n/neyronnye-seti-v-ekonomicheskom-analize-plyusy-i-minusy (дата обращения: 18.04.2024).

9. Артемьев В.Б. СУЭК - итоги 2017 года // Уголь. 2018. №3 (1104). URL: https://cyberleninka.ru/article/n/suek-itogi-2017-goda (дата обращения: 18.04.2024).

10. Артемьев В.Б. АО «СУЭК»: основные итоги работы в 2019 году // Уголь. 2020. №3 (1128). URL: https://cyberleninka.ru/article/n/ao-suek-osnovnye-itogi-raboty-v-2019-godu (дата обращения: 01.05.2024).

11. Грушина Н.В. Анализ экспорта угля в россии // Экономика и социум. 2015. №6-2 (19). URL: https://cyberleninka.ru/article/n/analiz-eksporta-uglya-v-rossii (дата обращения: 01.05.2024).

12. Горшенина Е.В. Инфляция // Экономические исследования. 2017. №4. URL: https://cyberleninka.ru/article/n/inflyatsiya (дата обращения: 01.05.2024).

13. Суворов А.В. Учет инфляции // Международный бухгалтерский учет. 2008. №4. URL: https://cyberleninka.ru/article/n/uchet-inflyatsii (дата обращения: 01.05.2024).

14. Астаева В.В., Балаева А.Ю. ИНФЛЯЦИЯ // Экономика и социум. 2015. №5-1 (18). URL: https://cyberleninka.ru/article/n/inflyatsiya-1 (дата обращения: 01.05.2024).

15. Кинтонова А.Ж., Ким Е. Оптимизация бизнес-процессов // Sciences of Europe. 2016. №9-4 (9). URL: https://cyberleninka.ru/article/n/optimizatsiya-biznes-protsessov-1 (дата обращения: 01.04.2024).

16. Тонерян Мкртыч Саркисович Исследование эффективности сезонных методов анализа временных рядов для прогнозирования объемов электропотребления // Инновации в науке. 2014. №29. URL: https://cyberleninka.ru/article/n/issledovanie-effektivnosti-sezonnyh-metodov-analiza-vremennyh-ryadov-dlya-prognozirovaniya-obemov-elektropotrebleniya (дата обращения: 25.05.2024).

17. Базилевский М.П. Исследование новых критериев для обнаружения автокорреляции остатков первого порядка в регрессионных моделях // Математика и математическое моделирование. 2018. №3. URL: https://cyberleninka.ru/article/n/issledovanie-novyh-kriteriev-dlya-obnaruzheniya-avtokorrelyatsii-ostatkov-pervogo-poryadka-v-regressionnyh-modelyah (дата обращения: 25.05.2024).

18. Матыцын Александр Владимирович Применение lstm-модели к прогнозированию cpi и уровня инфляции на примере россии // Вопросы инновационной экономики. 2022. №2. URL: https://cyberleninka.ru/article/n/primenenie-lstm-modeli-k-prognozirovaniyu-spi-i-urovnya-inflyatsii-na-primere-rossii (дата обращения: 25.05.2024).

19. Линдигрин Александр Николаевич Искусственные нейронные сети как основа глубинного обучения // Известия ТулГУ. Технические науки. 2019. №12. URL: https://cyberleninka.ru/article/n/iskusstvennye-neyronnye-seti-kak-osnova-glubinnogo-obucheniya (дата обращения: 28.05.2024).

20. Архипова А.А. Применение нейронных сетей в задаче прогнозирования финансовых временных рядов // Экономика и бизнес: теория

и практика. 2023. №6-1 (100). URL: https://cyberleninka.ru/article/n/primenenie-neyronnyh-setey-v-zadache-prognozirovaniya-finansovyh-vremennyh-ryadov (дата обращения: 03.06.2024).

21. Семёнов Евгений Дмитриевич, Брагинский Михаил Яковлевич, Тараканов Дмитрий Викторович, Назарова Инесса Леонидовна Нейросетевое прогнозирование входных параметров при добыче нефти // ВК. 2023. №4. URL: https://cyberleninka.ru/article/n/neyrosetevoe-prognozirovanie-vhodnyh-parametrov-pri-dobyche-nefti (дата обращения: 03.06.2024).

ПРИЛОЖЕНИЕ А
Алгоритм предобработки данных

```
import pandas as pd
import numpy as np
from pandas import read_csv, DataFrame
%matplotlib inline
import scipy.integrate as integrate
import scipy
from scipy import stats
from scipy.stats import norm
import scipy.stats as stats
from datetime import datetime
import statsmodels.formula.api as smf
import statsmodels.api as sm
import datetime as dt
import matplotlib.pyplot as plt
import math
import time
from sklearn.preprocessing import MinMaxScaler



dataset = '/content/CoalCSV (1).csv'
df = pd.read_csv(dataset)
df = df[['Coal']]
print(df)
print(df.info)

# Формирование обучающей выборки
# Определение размера обучающей выборки
size_train = math.ceil(len(df) * .85)
size_train

#Разделение дата сета на обучающу и тестовую выборки
train_data_df = df[:size_train].iloc[:,:1]
test_data_df = df[size_train:].iloc[:,:1]
```

```
print(train_data_df.shape, test_data_df.shape)

# Приведение данных из датафрейма в массив
train_data = train_data_df.Coal.values
test_data = test_data_df.Coal.values
# Изменения массива из 1D в 2D
train_data = np.reshape(train_data, (-1,1))
test_data = np.reshape(test_data, (-1,1))
train_data

# подтверждение стационарности
test = sm.tsa.adfuller(df['Coal'])
print('adf: ', test[0])
print('p-value: ', test[1])
if test[0] > test[1]:
  print("есть единичные корни, ряд не стационарен")
else:
  print("единичных корней нет, ряд стационарен")
df.describe()

# Создаем простой минмаксный скейлер для нормализации данных
scaler = MinMaxScaler(feature_range=(0,1))
# Тренируем скейлер и нормализуем обучающую выборку
train_data = scaler.fit_transform(train_data)

# Нормализуем тестовую выборку
test_data = scaler.transform(test_data)
print(test_data[:5])

# Подготавливаем данные к обучению
# Создаем массивы для входных данных и выходных
X_train = []
y_train = []
for i in range(6, len(train_data)):
  # В массив входных данных помещается показатели 6 предыдущих месяцев
```

```
  X_train.append(train_data[i-6:i, 0])
  # В массив выходных данных помещается показатель за текущей месяц
  y_train.append(train_data[i, 0])

# Переводим в numpy массивы, с которыми работает нейросеть
X_train = np.array(X_train)
y_train = np.array(y_train)
print(X_train[0])
print(y_train[0])

# То же проделываем с тестовой выборкой
X_test = []
y_test = []
for i in range(6, len(test_data)):
  X_test.append(test_data[i-6:i, 0])
  y_test.append(test_data[i, 0])

X_test = np.array(X_test)
y_test = np.array(y_test)
print(X_test[0])
print(y_test[0])
```

ПРИЛОЖЕНИЕ Б

Реализация и результаты моделирования RNN-модели нейронной сети

```
# Инициализация RNN нейронной сети
regressorRNN = Sequential()

# Добавляем входной слой RNN, устанавливаем размер входных данных 6
regressorRNN.add(SimpleRNN(units = 64,
            activation = "tanh",
            return_sequences = True,
            input_shape = (X_train.shape[1],1)))

# Добавляем Dropuot слой для предотвращения переобучения
regressorRNN.add(Dropout(0.2))

# Добавляем еще RNN слои
regressorRNN.add(SimpleRNN(units = 64,
            activation = "tanh",
            return_sequences = True))

regressorRNN.add(SimpleRNN(units = 64))

# Добавляем выходной слой с 1 нейронном
regressorRNN.add(Dense(units = 1, activation='relu'))

# Компилируем модель
regressorRNN.compile(optimizer = 'Adam',
          loss = 'mean_squared_error')

# Обучаем модель и засекаем время на обучение модели
time_start_fitRNN = time.time()
history = regressorRNN.fit(X_train, y_train, batch_size=16, epochs=50)
time_end_fitRNN = time.time()
regressorRNN.summary()
```

```
# Сохраняем из истории обучения показатели ошибки
lossRNN = history.history['loss']


# Строим график ошибки в процессе обучения
plt.plot(lossRNN)
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train'], loc = 'upper left')
plt.show()


# Выполняем предсказание на тестовой выборке
predictRNN_test = regressorRNN.predict(X_test)
# Формируем предсказание на еще один год
# Берем последнее значение из тестовой выборки
temp_x = X_test[-1:]
predictRNN = []
# В цикле пробегаемся по 13 месяцам, последний прошлого года и 12 следующего
for i in range(13):
    # Делаем предсказание на следующий месяц
    predictRNN.append(regressorRNN.predict(temp_x)[0, 0])
    # Изменяем данные на которых делаем предсказание
    # Откидываем самый ранний месяц из начала
    # И добавляем данные за предсказанный месяц в конец
    temp_x = np.append(temp_x[0][1:], predictRNN[-1]).reshape((1, 6))
predictRNN = np.array(predictRNN)


# Строим график для тестовой выборки и еще одного года
date = range(y_test.shape[0])
predict_date = range(y_test.shape[0]-1, y_test.shape[0] + predictRNN.shape[0]-1)
plt.plot(date, scaler.inverse_transform(y_test.reshape((y_test.shape[0], 1))))
plt.plot(date, scaler.inverse_transform(predictRNN_test.reshape((y_test.shape[0], 1))))
```

```
plt.plot(predict_date,
scaler.inverse_transform(predictRNN.reshape((predictRNN.shape[0], 1))))

plt.title('Тестирование модели RNN')

plt.ylabel('Объем добычи в тоннах')

plt.xlabel('Дата')

plt.legend(['Истинные значения теста','Предсказанные значения теста',
'Предсказание на следующий год'], loc = 'upper right')

plt.show()
```

ПРИЛОЖЕНИЕ В

Реализация и результаты моделирования LSTM-модели нейронной сети

```
# Инициализация LSTM нейронной сети
regressorLSTM = Sequential()

# Добавляем входной слой LSTM, устанавливаем разммер входных данных 6
regressorLSTM.add(LSTM(64,
                       return_sequences = True,
                       input_shape = (X_train.shape[1],1)))

# Добавляем Dropuot слой для предотвращения переобучения
regressorLSTM.add(Dropout(0.2))

# Добавляем еще LSTM слои
regressorLSTM.add(LSTM(64,
                       return_sequences = False))

# Добавляем дополнительный простой слой
regressorLSTM.add(Dense(256))

# Добавляем выходной слой с 1 нейронном
regressorLSTM.add(Dense(1))

# Компилируем модель
regressorLSTM.compile(optimizer = 'adam',
                      loss = 'mean_squared_error')

# Обучаем модель и засекаем время на обучение модели
time_start_fitLSTM = time.time()
history = regressorLSTM.fit(X_train,
                  y_train,
                  batch_size = 16,
                  epochs = 200)
time_end_fitLSTM = time.time()
```

```
regressorLSTM.summary()


# Сохраняем из истории обучения показатели ошибки
lossLSTM = history.history['loss']


# Строим график ошибки в процессе обучения
plt.plot(lossLSTM)
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train'], loc = 'upper left')
plt.show()


# Выполняем предсказание на тестовой выборке
predictLSTM_test = regressorLSTM.predict(X_test)


# Формируем предсказание на еще один год
temp_x = X_test[-1:]
predictLSTM = []
for i in range(13):
    predictLSTM.append(regressorLSTM.predict(temp_x)[0, 0])
    temp_x = np.append(temp_x[0][1:], predictLSTM[-1]).reshape((1, 6))
predictLSTM = np.array(predictLSTM)


# Строим график для тестовой выборки и еще одного года
date = range(y_test.shape[0])
predict_date    =       range(y_test.shape[0]-1,    y_test.shape[0]    +
predictLSTM.shape[0]-1)
plt.plot(date,
scaler.inverse_transform(y_test.reshape((y_test.shape[0], 1))))
plt.plot(date,
scaler.inverse_transform(predictLSTM_test.reshape((y_test.shape[0],
1))))
plt.plot(predict_date,
scaler.inverse_transform(predictLSTM.reshape((predictLSTM.shape[0],
1))))
plt.title('Тестирование модели LSTM')
```

```
plt.ylabel('Объем добычи в тоннах')

plt.xlabel('Дата')

plt.legend(['Истинные значения теста','Предсказанные значения теста',
'Предсказание на следующий год'], loc = 'upper right')

plt.show()
```

ПРИЛОЖЕНИЕ В

Реализация и результаты моделирования GRU-модели нейронной сети

```
# Инициализация нейронной сети
regressorGRU = Sequential()

# Добавляем входной слой GRU, устанавливаем разммер входных данных 6
regressorGRU.add(GRU(units=64,
          return_sequences=True,
          input_shape=(X_train.shape[1],1),
          activation='tanh'))

# Добавляем Dropuot слой для предотвращения переобучения
regressorGRU.add(Dropout(0.2))

# Добавляем еще GRU слои
regressorGRU.add(GRU(units=64,
          return_sequences=True,
          activation='tanh'))

regressorGRU.add(GRU(units=64,
          activation='tanh'))

# Добавляем выходной слой с 1 нейронном
regressorGRU.add(Dense(units=1,
          activation='relu'))

# Компилируем модель
regressorGRU.compile(optimizer='adam',
          loss='mean_squared_error')

# Обучаем модель и засекаем время на обучение модели
time_start_fitGRU = time.time()
history = regressorGRU.fit(X_train,y_train,epochs=100,batch_size=16)
time_end_fitGRU = time.time()
regressorGRU.summary()
```

```
# Сохраняем из истории обучения показатели ошибки
lossGRU = history.history['loss']

# Строим график ошибки в процессе обучения
plt.plot(lossGRU)
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train'], loc = 'upper left')
plt.show()

# Выполняем предсказание на тестовой выборке
predictGRU_test = regressorGRU.predict(X_test)

# Формируем предсказание на еще один год
temp_x = X_test[-1:]
predictGRU = []
for i in range(13):
    predictGRU.append(regressorGRU.predict(temp_x)[0, 0])
    temp_x = np.append(temp_x[0][1:], predictGRU[-1]).reshape((1, 6))
predictGRU = np.array(predictGRU)

# Строим график для тестовой выборки и еще одного года
date = range(y_test.shape[0])
predict_date    =       range(y_test.shape[0]-1,    y_test.shape[0]    +
predictGRU.shape[0]-1)
plt.plot(date,
scaler.inverse_transform(y_test.reshape((y_test.shape[0], 1))))
plt.plot(date,
scaler.inverse_transform(predictGRU_test.reshape((y_test.shape[0],
1))))
plt.plot(predict_date,
scaler.inverse_transform(predictGRU.reshape((predictGRU.shape[0], 1))))
plt.title('Тестирование модели GRU')
plt.ylabel('Объем добычи в тыс. тонн')
plt.xlabel('Дата, по месяцам')
```

```
plt.legend(['Истинные значения теста','Предсказанные значения теста',
'Предсказание на следующий год'], loc = 'upper right')
plt.show()
```

ПРИЛОЖЕНИЕ Г

Сравнительные графики результатов предсказания моделей нейронных сетей. Статистические сведения о результате работы моделей.

```
# Строим сравнительные графики на тестовой выборке для моделей
# Определяем даты
date = range(y_test.shape[0])
predict_date =  range(y_test.shape[0]-1, y_test.shape[0]
+ predictRNN.shape[0]-1)

# Выставляем размер графика
figure = plt.figure()
figure.set_figwidth(13)
figure.set_figheight(7)

# Строим график для истинных значений тестовой выборки
plt.plot(date,
        scaler.inverse_transform(y_test.reshape((y_test.shape[0], 1))),
color = 'darkgrey',
         label='Истинные значения теста')

# Строим график предсказаний RNN на тестовой выборке
plt.plot(date,
          scaler.inverse_transform(predictRNN_test.reshape((y_test.shap
e[0], 1))),
         color='firebrick', label='Предсказание модели RNN')
# Достраиваем предсказание за следующий год для RNN
plt.plot(predict_date,
          scaler.inverse_transform(predictRNN.reshape((predictRNN.shape
[0], 1))),
         color='firebrick')

# Строим график предсказаний LSTM на тестовой выборке
plt.plot(date,
          scaler.inverse_transform(predictLSTM_test.reshape((y_test.sha
pe[0], 1))),
         color='blue', label='Предсказание модели LSTM')
```

```
# Достраиваем предсказание за следующий год для LSTM
plt.plot(predict_date,
         scaler.inverse_transform(predictLSTM.reshape((predictLSTM.sha
pe[0], 1))),
         color='blue')

# Строим график предсказаний GRU на тестовой выборке
plt.plot(date,
         scaler.inverse_transform(predictGRU_test.reshape((y_test.shap
e[0], 1))),
         color='limegreen', label='Предсказание модели GRU')
# Достраиваем предсказание за следующий год для GRU
plt.plot(predict_date,
         scaler.inverse_transform(predictGRU.reshape((predictGRU.shape
[0], 1))),
         color='limegreen')

plt.title('Сравнение моделей')
plt.ylabel('Объем добычи в тоннах')
plt.xlabel('Дата')
plt.legend(loc = 'upper right')
plt.show()

# Строим сравнительные графики на следующий год для моделей
# Определяем даты
predict_date =  range(13)

# Выставляем размер графика
figure = plt.figure()
figure.set_figwidth(13)
figure.set_figheight(7)

# Строим график предсказания за следующий год для RNN
plt.plot(predict_date,
scaler.inverse_transform(predictRNN.reshape((predictRNN.shape[0], 1))),
color='firebrick', label='Предсказание модели RNN')
```

```
# Строим график предсказания за следующий год для LSTM
plt.plot(predict_date,
scaler.inverse_transform(predictLSTM.reshape((predictLSTM.shape[0],
1))), color='blue', label='Предсказание модели LSTM')


# Строим график предсказания за следующий год для GRU
plt.plot(predict_date,
scaler.inverse_transform(predictGRU.reshape((predictGRU.shape[0], 1))),
color='limegreen', label='Предсказание модели GRU')


plt.title('Сравнение предсказания моделей на следующий год')
plt.ylabel('Объем добычи в тоннах')
plt.xlabel('Дата')
plt.legend(loc = 'upper right')
plt.show()


print(f"Время обучение модели RNN:{time_end_fitRNN - time_start_fitRNN}
секунд")
print(f"Время обучение модели LSTM:{time_end_fitLSTM -
time_start_fitLSTM} секунд")
print(f"Время обучение модели GRU:{time_end_fitGRU - time_start_fitGRU}
секунд")


# Строим график сравнения функции ошибки для каждой модели
# Из-за разного количества эпох график является не совсем иллюстративным
plt.plot(lossRNN)
plt.plot(lossLSTM)
plt.plot(lossGRU)
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['RNN', 'LSTM', 'GRU'], loc = 'upper left')
plt.show()


print(f"Ошибка модели RNN:{lossRNN[-1]}")
print(f"Ошибка модели LSTM:{lossLSTM[-1]}")
print(f"Ошибка модели GRU:{lossGRU[-1]}")
```

```
# Строим датафрейм из предсказанных данных на следующий год для каждой модели
df_result = pd.DataFrame({"RNN_predict": scaler.inverse_transform(predictRNN.reshape((predictRNN.shape[0], 1))).reshape(predictRNN.shape[0]),
                          "LSTM_predict": scaler.inverse_transform(predictLSTM.reshape((predictLSTM.shape[0], 1))).reshape(predictLSTM.shape[0]),
                          "GRU_predict": scaler.inverse_transform(predictGRU.reshape((predictGRU.shape[0], 1))).reshape(predictGRU.shape[0])})
df_result.round(2)
```

ПРИЛОЖЕНИЕ Д

Содержание письменных обращений к АО «СУЭК» и Министерству угольной промышленности Кузбасса

Кирчевская Полина Вячеславовна

Полина Кирчевская polyiikon@yandex.ru 20 мая в 3:13
office@suek.ru ›

suT8k9LKKoA.
jpg

Добрый день, уважаемая компания СУЭК.

Меня зовут Кирчевская Полина, мне 21 год, я бакалавр 4-го курса института СПбГУАП по специальности "Прикладная информатика". В настоящее время я веду написание дипломной работы ВКР на тему "Решение задачи прибыли промышленного предприятия с помощью нейронных сетей", где пишу алгоритм аналитики и прогноза данных прибыли угледобывающего предприятия "СУЭК" за период 2019 - 2024 года с прогнозированием объемов добычи на 1.5 года вперед. В данную дипломную работу также входит аналитика и прогнозирование уровня инфляции сырьевого сектора РФ и влияние найденных макроэкономических показателей на прогнозируемую динамику стоимости 1(одной) тонны добытого угля (как закрытым так и открытым методом).

В настоящее время с интернете содержится крайне мало информации о помесячных объемов добычи угля вашим предприятием. Прошу вас посодействовать в написании моей выпускной квалификационной работе в предоставлении данных помесячных (можно чаще) объемов добычи угля с 2019 по 2024 год для проведения статистического анализа и построения прогнозов.
Для обучения моей модели нейронной сети требуется достаточно большой объем данных. который весьма затруднительно найти в свободном плавании по интернету.

Для подтверждения личности прикрепляю к письму фото своего продленного студенческого билета

Прошу вашей помощи
С уважением, Кирчевская Полина

P.S. Используйте эту почту для обратной связи, пожалуйста

ПОЛИНА КИРЧЕВСКАЯ

Полина Кирчевская polyiikon@yandex.ru 21 мая в 1:19
dep_tek@ako.ru ›

suT8k9LKKoA.
jpg

Добрый день, уважаемое министерство угольной промышленности.

Меня зовут Кирчевская Полина, мне 21 год, я бакалавр 4-го курса института СПбГУАП по специальности "Прикладная информатика". В настоящее время я веду написание дипломной работы ВКР на тему "Решение задачи прибыли промышленного предприятия с помощью нейронных сетей", где пишу алгоритм аналитики и прогноза данных прибыли угледобывающего предприятия за период 2019 - 2024 года с прогнозированием объемов добычи на 1.5 года вперед. В данную дипломную работу также входит аналитика и прогнозирование уровня инфляции сырьевого сектора РФ и влияние найденных макроэкономических показателей на прогнозируемую динамику стоимости 1(одной) тонны добытого угля (как закрытым так и открытым методом).

В настоящее время с интернете содержится крайне мало информации о помесячных объемов добычи угля вашим предприятием. Прошу вас посодействовать в написании моей выпускной квалификационной работе в предоставлении данных помесячных (можно чаще) объемов добычи угля с 2019 по 2024 год для проведения статистического анализа и построения прогнозов.
Для обучения моей модели нейронной сети требуется достаточно большой объем данных. который весьма затруднительно найти в свободном плавании по интернету.

Для подтверждения личности прикрепляю к письму фото своего продленного студенческого билета

Прошу вашей помощи
С уважением, Кирчевская Полина

P.S. Используйте эту почту для обратной связи, пожалуйста